GÉNÉRATEUR DE TEXTES ALÉATOIRES AUTOMATIQUEMENT LEMMATISÉS

Mémoire présenté comme exigence partielle
de la maîtrise en technologie de l'information

Par Jean-Philippe Dionne

Août 2024

RÉSUMÉ

La lemmatisation d'un texte consiste à réduire chaque mot contenu dans celui-ci à sa forme de base. Cette opération revient à grouper tous les verbes conjugués sous leur forme infinitive, tous les adjectifs sous leur forme masculin singulier, et tous les noms sous leur forme au singulier. C'est d'ailleurs cette forme de base, appelée « lemme », qu'on retrouve dans les entrées de dictionnaires et non les formes dites fléchies. La lemmatisation d'un texte s'effectue dans plusieurs contextes tels que par exemple la recherche de documents par mot-clé, l'analyse du style d'un auteur ou la quantification de la richesse lexicale d'un texte écrit ou oral, ou comme première étape visant à faciliter la traduction automatique de documents.

La lemmatisation d'un mot en particulier peut dans la plupart des cas s'effectuer en traitant ce mot seul sans tenir compte du contexte où il est utilisé ou des mots qui l'entourent. Cependant, une lemmatisation appropriée exige de pouvoir distinguer les homographes, c'est-à-dire des mots dont la graphie est la même mais dont le sens et ultimement le lemme et possiblement la classe grammaticale diffèrent. Les homographes ne peuvent être lemmatisés de façon appropriée sans tenir compte du contexte de leur utilisation.

Vu la complexité de la désambiguïsation des homographes, il est critique de pouvoir évaluer la performance des algorithmes de lemmatisation automatique disponibles. Mais une telle évaluation requiert la mise en place d'un étalon de référence avec lequel comparer les résultats de chaque algorithme. Mais hélas, comme il n'existe encore aucun algorithme de lemmatisation parfait, cet étalon ne peut se bâtir que par une analyse manuelle d'un texte, une tâche laborieuse.

Afin de pallier ce besoin de lemmatisation manuelle, le projet actuel a permis de générer des textes aléatoires en français automatiquement lemmatisés dont la précision en termes de lemmatisation est, par défaut, parfaite. Ces textes générés par ordinateur ne sont donc pas lemmatisés *après coup*, mais plutôt au moment de leur construction.

Une telle génération de textes aléatoires automatiquement lemmatisés a requis la mise en place de banques de données de lemmes ainsi que l'emploi de règles précises pour générer les formes fléchies de ces lemmes. Une approche innovante basée sur l'utilisation d'un *corpus de référence* a été adoptée pour générer ces banques de mots. L'apprentissage machine appliqué au traitement syntaxique des homographes a été introduit pour assurer un meilleur accord entre le lexique du corpus de référence et celui utilisé pour les phrases aléatoires.

L'outil développé dans le cadre de cette recherche a donc permis de créer des phrases lemmatisées dont la structure correspond aux normes de la langue française. En revanche, il n'a pas été possible de générer des textes porteurs de sens, car un tel objectif aurait été trop ambitieux. Une fois ces textes générés, on a procédé à l'évaluation d'outils de lemmatisation sur la base de ces textes ainsi que certains autres, afin d'en déterminer la fiabilité.

Mots-clés : lemmatisation, homographes, désambiguïsation, apprentissage machine

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTE DES EXTRAITS DE CODE

TABLE DES MATIERES

1. INTRODUCTION

Le présent travail s'inscrit dans le champ général du traitement automatique des langues naturelles (TALN). Tel que décrit par Yvon (2010), le TALN est « un domaine de recherche visant à modéliser et reproduire, à l'aide de machines, la capacité humaine à produire et à comprendre des énoncés linguistiques dans des buts de communication ». Fuchs et Habert (2004) insistent quant à eux sur le besoin de concevoir des logiciels informatiques pour effectuer un tel traitement, mariant ainsi deux grands domaines de recherche, soit la linguistique et l'informatique. Ils opposent ainsi le langage naturel aux langages formels de la logique mathématique. Les langages formels informatiques par exemple, se distinguent des langages naturels par leur absence d'ambiguïté. En effet, un programme « Java » fournit les mêmes résultats en sortie, pourvu bien sûr que les données en entrées soient les mêmes, quelle que soit la machine sur laquelle il est exécuté. Cette absence d'ambiguïté caractéristique des langages informatiques est essentielle pour l'universalité des applications d'algorithmes et en bout de ligne l'acceptation de ces outils pour gérer nos activités humaines.

Au contraire de ces langages formels, le langage naturel comporte son lot d'ambiguïtés. À l'oral par exemple, les deux phrases suivantes sont prononcées de la même façon. Mais leur sens est complètement différent :

- J'ai besoin d'un verre.
- J'ai besoin d'un ver.

On est ici en présence d'homophones, mots aux sens différents se prononçant de la même façon. D'autres ambiguïtés du langage naturel portent sur la sémantique, comme pour les deux exemples suivants, retrouvés sur Wikipedia (2023a) sous la rubrique « ambiguïté »:

- Je t'embrasse, et Yves aussi.
- Un homme peint la nuit.

Dans le premier cas, il n'est pas clair si j'embrasse aussi Yves, ou si c'est plutôt Yves qui embrasse lui aussi la même personne que j'embrasse. Dans le deuxième exemple, il n'est pas clair si l'homme en question peint une scène représentant la nuit, ou si cette phrase ne fait que préciser que l'homme peint pendant la nuit, sans préciser le sujet de son œuvre.

Bien que l'ambiguïté du langage naturel puisse être perçue comme problématique, celle-ci est parfois au contraire recherchée pour créer un effet particulier, par exemple pour introduire une touche humoristique. Lee (2014) par exemple, dans sa thèse portant sur le slogan publicitaire, précise que le « recours à des mots à sens multiples, ambigus, constitue l'un des actes majeurs du slogan publicitaire. » Il cite plusieurs exemples dont : « L'épargne qui attire l'intérêt ». En effet, le mot « intérêt » ici peut être interprété à la fois comme un terme technique du domaine bancaire, ou encore pour démontrer que l'attention est attirée.

Mais dans le présent travail de recherche, il n'est ni question de l'ambiguïté homophonique citée plus haut, ni de l'ambiguïté sémantique chère au domaine publicitaire, mais plutôt de l'ambiguïté homographique. Des homographes sont des mots écrits exactement de la même manière, mais ne signifiant pas la même chose. Des homographes peuvent appartenir à des classes grammaticales différentes :

- Je bois de l'eau. (le mot « bois » est un verbe)
- Je traverse le bois. (le mot « bois » est un nom commun)

Ils peuvent aussi appartenir à la même classe :

- Je parle à mon fils. (le mot « fils » est un nom commun au singulier, synonyme d'enfant de sexe masculin)
- Les fils électriques s'entremêlent. (le mot « fils » est nom commun au pluriel, signifiant un composant électrotechnique)

L'identification et la désambiguïsation des homographes est essentielle lors de la lemmatisation de textes. La lemmatisation d'un texte consiste à réduire chaque mot contenu dans celui-ci à sa forme de base. Cette opération revient à grouper tous les verbes conjugués sous leur forme infinitive, tous les adjectifs sous leur forme masculin singulier, et tous les noms sous leur forme au singulier. C'est d'ailleurs cette forme de base, appelée forme canonique ou « lemme », qu'on retrouve dans les entrées de dictionnaires et non les formes dites fléchies. Tel que le mentionne Liu et al. (2012), la lemmatisation réduit ainsi le nombre de termes distincts dans un texte, ce qui permet d'en diminuer la complexité pour analyse subséquente.

Gross (2004) insiste justement sur l'imposant défi de mettre au point des lemmatiseurs pouvant automatiquement reconnaitre la forme ou la classe des mots identifiés dans un texte, considérant l'ambiguïté inhérente à la langue écrite, par le biais des homographes. Bien que la langue française ne soit pas unique à cet égard, elle comporte un très grand nombre d'homographes. Ce phénomène a été exacerbé par la chute de consonnes de plusieurs mots au cours des siècles, ayant emmené des mots autrefois distincts, à adopter la même graphie. C'est ce que Glikman et Perret (2008) définissent comme l'érosion phonétique. La très grande fréquence d'homographes fait en sorte qu'aucun outil informatique actuel ne peut obtenir un score parfait quand vient le temps de lemmatiser un texte en français.

C'est dans ce contexte que le présent travail cherche à faciliter la quantification de la performance de lemmatiseurs de la langue française existants, en leur fournissant des textes de référence automatiquement lemmatisés, qui serviront d'étalon. En effet, l'évaluation d'outils de lemmatisation est une tâche ardue, puisqu'elle requiert normalement une vérification du bon classement de chaque homographe, une opération manuelle fastidieuse. L'outil développé ici permet donc une évaluation bien plus rapide et plus efficace, ainsi qu'une comparaison facilitée entre les différents outils.

1.1. Objectifs de la recherche

L'objectif principal de ce travail de recherche est de mettre au point un outil permettant de générer de façon aléatoire des textes de longueur variable automatiquement lemmatisés, sans qu'aucun effort manuel de lemmatisation soit requis. Ces textes, étant lemmatisés automatiquement au moment de leur création, plutôt qu'après coup, offrent par défaut une lemmatisation parfaite.

De tels textes parfaitement lemmatisés peuvent servir d'étalon (« *gold standard* ») pour l'évaluation de la performance d'outils de lemmatisation existants, considérant qu'aucun de ceux-ci ne réussit à accomplir la tâche parfaitement. Encore de nos jours, malgré l'avancement des connaissances et de la puissance des outils informatiques, une lemmatisation parfaite d'un texte quelconque ne peut s'effectuer que manuellement, un effort laborieux pour de longs textes.

Ces textes générés aléatoirement s'inspirent d'un corpus de référence choisi par l'utilisateur. Au besoin, ce corpus de référence peut être représentatif d'un domaine de connaissances en particulier, par exemple la médecine, l'architecture ou la littérature d'une époque ou d'un style donnés. Ainsi, on peut juger la performance d'outils de lemmatisation dans des contextes précis, ce qui peut s'avérer très pertinent dans les cas où justement, l'outil de lemmatisation doit être appliqué dans l'un de ces contextes.

Les objectifs principaux de ce travail de recherche peuvent être résumés comme suit :

- Générer aléatoirement des textes automatiquement lemmatisés aussi longs que nécessaire pouvant servir d'étalons de référence pour l'évaluation d'algorithmes de lemmatisation.
- Mettre l'emphase sur la difficulté inhérente de lemmatiser les homographes.
- Permettre un meilleur traitement statistique de l'évaluation des algorithmes de lemmatisation en les confrontant à un grand nombre de textes générés aléatoirement.
- Permettre une évaluation de la rapidité des algorithmes de lemmatisation en les confrontant à de très longs textes.

1.1.1. Modalités du projet

D'emblée, certaines décisions ont été prises concernant la façon de mener le projet à bien. Les modalités suivantes caractérisent donc le projet :

- L'algorithme de génération de textes aléatoires est conçu dans le langage Java.
- Des banques de données lexicales sont construites en fichiers texte, contenant des listes de mots selon leurs catégories lexicales (noms communs, déterminants, adjectifs, etc.)
- Un sous-algorithme permet de générer un très grand nombre de verbes conjugués.
- La version de l'algorithme fournie en livrable n'a pas la prétention de pouvoir répliquer n'importe quelle phrase de la langue française; seules certaines structures type simples (arbres syntaxiques) sont reproduites.
- L'algorithme peut être ajusté en fonction des besoins à l'aide de paramètres fournis en entrée, comme la proportion des temps de verbe par exemple.
- En sortie, l'algorithme fournit le texte généré, le lemme de chacun des mots et son rôle dans la phrase (étiquetage morpho-syntaxique). Il fournit aussi des statistiques globales qu'on peut comparer avec celles fournies par les algorithmes de lemmatisation à évaluer.

1.1.2. Limites de l'étude

L'outil développé ne bâtit des phrases que selon des structures simples, par exemple « déterminant-nom commun-verbe-déterminant-nom commun ». De plus nombreuses variations de structures de phrases pourront être ajoutées par la suite, au-delà de la complétion du projet.

Il va de soi que les textes générés par l'algorithme n'ont pas de valeur sémantique. On peut se retrouver par exemple avec une phrase complètement vide de sens telle que « Le camion endormi mange la diversification ».

La portée du présent inclut une évaluation sommaire d'outils de lemmatisation existants en comparant leurs performances, mais aucune analyse plus approfondie n'est faite sur le fonctionnement de leurs algorithmes respectifs.

1.2. Motivation et pertinence du projet

La lemmatisation d'un texte s'effectue dans plusieurs contextes tels que par exemple la recherche de documents par mot-clé. En effet, une recherche visant à identifier des documents pertinents contenant le verbe « traiter » sera plus efficace s'il est possible d'extraire des documents analysés toutes les formes conjuguées du verbe « traiter ». De la même façon, une recherche Internet sur les chevaux aura plus de succès si le moteur de recherche permet de repérer autant les occurrences de « cheval » au singulier, que « chevaux » au pluriel.

La lemmatisation peut aussi servir à analyser le style d'un auteur ou la quantification de la richesse lexicale d'un texte écrit ou oral (Ovtcharov et al., 2006). La richesse lexicale dénote le nombre

total de mots distincts, ou le ratio de mots distincts sur le nombre total de mots utilisés dans un texte. Un auteur peut par exemple faire appel à un outil de lemmatisation pour s'assurer d'intégrer une grande variété dans le choix des mots qui composent son texte, évitant la répétition de certains verbes, noms ou adjectifs dont il aurait abusé. Il serait en effet difficile à un auteur de déterminer le nombre de fois qu'il a utilisé le verbe « faire » dans son texte, s'il devait en répertorier toutes les formes l'une après l'autre (« fais », « faisais », « ferions », etc.) à l'aide de recherches manuelles dans un logiciel de traitement de texte.

La lemmatisation peut aussi servir de première étape visant à faciliter la traduction automatique de documents (Mahmoud, 2002), puisqu'il est plus facile de trouver les traductions de lemmes que de formes fléchies, les formes fléchies variant énormément d'une langue à une autre. De plus, un outil de lemmatisation permet généralement aussi d'étiqueter les mots, ce qui permet d'en déterminer la fonction dans la phrase, incluant leur classe grammaticale et autres paramètres pertinents. Là encore, l'opération de traduction s'en retrouve grandement simplifiée.

Hélas, la lemmatisation peut s'avérer un exercice très fastidieux, surtout lorsqu'il est question de longs textes. Heureusement, il existe déjà plusieurs algorithmes informatiques de lemmatisation dont certains sont même disponibles gratuitement sur le web. Mais leur précision n'est pas toujours au rendez-vous. Certains chercheurs en ont exploré l'efficacité dans différentes langues et pour des textes de différents domaines. Par exemple, en anglais pour les termes biomédicaux (Liu et al., 2012), en allemand général (Perera & Witte, 2005), et bien d'autres.

La lemmatisation d'un mot en particulier peut dans la plupart des cas s'effectuer en traitant ce mot seul sans tenir compte du contexte où il est utilisé ou des mots qui l'entourent. Cependant, une lemmatisation appropriée exige de pouvoir distinguer les homographes, c'est-à-dire des mots dont la graphie est la même mais dont le sens et ultimement le lemme diffèrent. Comme exemples d'homographes en français, on retrouve « fils », descendant immédiat de sexe masculin ou encore le pluriel de « fil », « est », du verbe être ou élément de la rose des vents, « Rose », le prénom, le nom commun ou l'adjectif, ainsi que des milliers d'autres. Les homographes ne peuvent être lemmatisés de façon appropriée sans tenir compte du contexte de leur utilisation.

Le traitement des homographes s'avère complexe et plusieurs chercheurs s'y sont penchés, encore là, pour plusieurs langues. Hein (1990) mentionne qu'environ 85% des homographes suédois proviennent de classes de mots différents, par exemple un nom et un verbe. Les 15% restants concernent des mots d'une même classe, par exemple deux verbes. Elle explique aussi l'origine de tels homographes ainsi que certaines stratégies pour tenter de les distinguer selon qu'ils appartiennent ou non à des classes de mots différents. Une approche semblable est nécessaire en français.

Vu la complexité de la tâche, surtout considérant le défi du traitement des homographes, il est critique de pouvoir évaluer la performance des algorithmes de lemmatisation automatique disponibles. Mais une telle évaluation requiert la mise en place d'un « étalon de référence » ou « étalon doré » (« *gold standard* » en anglais) avec lequel on peut comparer les résultats de chaque algorithme. Mais hélas, comme il n'existe encore aucun algorithme de lemmatisation parfait, cet étalon doré ne peut se bâtir que par une analyse manuelle d'un texte, une tâche laborieuse. Liu et al. (2012) a toutefois suggéré une « norme argentée » (« *silver standard* ») générée automatiquement en se limitant aux analyses concordantes de plusieurs outils de lemmatisation utilisés en parallèle. Cependant, cette norme argentée laisse de côté les mots n'ayant pas été lemmatisés de la même façon par les différents outils. Il faut ensuite procéder manuellement pour lemmatiser les mots ayant échappé à cette norme argentée. On voit donc qu'il est inévitable de devoir effectuer au moins un certain niveau de lemmatisation manuelle pour établir un véritable étalon de référence lors de l'analyse d'un texte ou d'un corpus en particulier.

Afin de pallier ce besoin de lemmatisation manuelle, le projet actuel a pour but la génération aléatoire de textes en français automatiquement lemmatisés dont la précision en termes de lemmatisation est, par défaut, parfaite. Plus particulièrement, l'outil de génération de textes en question met l'emphase sur les homographes. L'outil en inclut plusieurs dans les textes aléatoires, accompagnés d'indices qui permettent aux algorithmes de lemmatisation de les distinguer entre eux, soit par des marques syntaxiques ou par la présence de cooccurrences pertinentes.

Une telle génération de textes aléatoires automatiquement lemmatisés requiert la mise en place de banques de données de lemmes ainsi que l'emploi de règles précises pour générer les formes fléchies de ces lemmes. Mais surtout, on doit créer des phrases dont la structure correspond aux normes de la langue française. Il n'est pas question ici de générer des textes porteurs de sens, car un tel objectif serait trop ambitieux. Bien au contraire, il faut s'attendre à ce que l'outil génère des phrases totalement dénuées de sens, mais dont la grammaire respecte toutefois toutes les règles de la langue française, ce qui est suffisant pour le besoin présent.

Ces textes générés par ordinateur ne sont donc pas lemmatisés « après coup », mais plutôt au moment de leur construction. Pour y arriver, on puise dans des banques de données lexicales qui comprennent des mots typiquement associés aux variations des homographes ainsi qu'en imitant certains modèles simples de structures de phrases. De nombreux homographes sont donc naturellement incorporés aux textes. La portée de ce projet inclut aussi l'évaluation d'outils de lemmatisation existants sur la base de ces textes.

1.3. Défis inhérents au travail de recherche

La langue française, tout comme toute autre langue, comporte un très grand lexique et son analyse et son traitement demandent donc de composer avec de larges banques de données. Un des défis inhérents à ce travail fut donc de créer de toutes pièces ces banques de données, par exemple des listes de mots de différentes classes grammaticales avec leurs paramètres, une liste quasi exhaustive des verbes de la langue française, ainsi que des tableaux de conjugaison. À ces données de base se rajoutent d'autres informations telles qu'une liste de mots avec « h » aspiré, puisque ceux-ci ne font pas appel à l'apostrophe, une liste d'adjectifs s'utilisant devant le nom (« antéposés »), etc. Les banques de données générées pour ce projet sont loin d'être exhaustives, mais ont tout de même permis l'analyse détaillée des corpus de référence choisis.

Le traitement de textes par ordinateur requiert que ceux-ci soient facilement lisibles et interprétables par les algorithmes. Il a donc fallu transférer des fichiers de départ en format Microsoft Word ou PDF en versions « texte », ce qui a demandé un certain nettoyage manuel. Et comme les banques de données utilisées sont plutôt larges et que les corpus de référence choisis comportent près de 200 000 mots, un autre défi a été de mettre au point des algorithmes rapides et efficaces assurant un temps d'exécution raisonnable. Une exécution rapide permet un développement et des opérations de débogage rapides.

Une autre difficulté a été de bâtir une structure pour toutes les données donnant toute la flexibilité nécessaire permettant d'emmagasiner, pour chaque mot du corpus, tous les homographes y correspondant, ainsi que tous les paramètres pour chaque mot fléchi ou lemme, par exemple son genre et son nombre, le cas échéant. Toutes ces informations ont dû être emmagasinées de façon à rendre leur rappel facile et rapide pour exécuter diverses tâches, telles que l'observation des cooccurrences et la désambiguïsation des homographes.

La désambiguïsation des homographes en soi a représenté un grand défi, car il a fallu identifier un certain nombre de caractéristiques des classes grammaticales, pour que l'algorithme assigne la classe la plus probable à chaque homographe. Des notions d'apprentissage machine ont été mises en application pour optimiser le résultat. La régression logistique s'est avérée l'approche la plus pertinente, mais la sensibilité de certains algorithmes aux données fournies en entrée a été

problématique par moments. En particulier, on a proposé en premier lieu un algorithme de classification évitant tout traitement manuel, dont la performance est limitée, mais dont la pertinence demeure, considérant le temps gagné. Des algorithmes plus avancés basés sur un entrainement manuel ont aussi été développés, ce qui a demandé davantage d'efforts.

Une autre grande difficulté, qui s'est traduite par une portée limitée au projet, fut celle de choisir des structures de phrases types de la langue française. Le but du projet n'était pas de reproduire une grande variété de constructions de phrases, mais plutôt de favoriser l'apparition d'homographes dans des phrases bien structurées selon les codes de la langue. En ce sens, malgré cette limitation dans les structures de phrase, le projet a tout de même atteint son but.

Finalement, bien que ce ne soit pas là un objectif principal du projet, un effort a été fait pour tenter de donner un minimum de sens aux phrases générées aléatoirement à la deuxième étape du projet. L'approche pour y arriver a consisté à faire apparaitre autour de chaque mot des mots situés aux alentours de ce même mot au sein du corpus de référence. Ces « cooccurrences » ont donc d'abord dû être identifiées, et des choix ont dû être faits pour déterminer les paramètres requis pour automatiser le processus. Ces cooccurrences ont le mérite potentiel de fournir des indices à des lemmatiseurs pour désambiguïser les homographes. Mais malheureusement, les phrases générées aléatoirement, tel qu'anticipé, n'ont que très rarement résulté en des phrases ayant une quelconque valeur sémantique. Mais là encore, cela n'a pas nui à l'objectif premier qui était de fournir des textes automatiquement lemmatisés dans le but d'évaluer des outils de lemmatisation existants. Toute valeur sémantique n'aurait représenté qu'un bonus.

1.4. Plan du rapport

Ce travail de recherche se développe en trois grandes étapes : d'abord la construction de banques de données lexicales, puis la génération de textes aléatoires automatiquement lemmatisés, en utilisant les banques de données créées lors de la première étape, et puis finalement l'évaluation d'outils de lemmatisation existants sur la base de ces textes.

A l'Étape 1, on crée une liste de lemmes qui sert par la suite à générer les textes aléatoires à l'Étape 2. Cette liste comprend des lemmes pour les classes grammaticales les plus importantes (verbes, noms communs, adjectifs et adverbes) qui permettent par la suite de construire toutes sortes de phrases réalistes. Cette liste de lemmes se bâtit grâce au développement et à l'utilisation d'un lemmatiseur qui extrait cette information d'un corpus de référence. Il peut paraître contradictoire ou paradoxal de concevoir un outil de lemmatisation dans le but ultime d'évaluer d'autres outils de lemmatisation. Mais il faut considérer que l'outil de lemmatisation créé pour cette première étape doit tout simplement pouvoir générer une grande liste de lemmes (verbes, noms communs, adjectifs et adverbes), sans qu'il soit nécessaire que ceux-ci soient exactement ceux correspondant au texte.

A l'Étape 1, on crée aussi pour chaque verbe, nom commun, adjectif et adverbe, une liste de cooccurrences, groupées par classe grammaticale. Ces cooccurrences permettent de générer des textes avec un minimum de sens à l'Étape 2. Aussi, on calcule la longueur moyenne des phrases et leur distribution statistique, ce qui encore là, sert de balise pour l'Étape 2.

L'Étape 2 représente quant à elle le cœur de ce projet, car l'Étape 1 ne sert qu'à fournir le matériel de base nécessaire pour effectuer l'Étape 2. À cette étape, un certain nombre de structures de phrases typiques ont été définies. Une analyse même rapide de tout texte en français nous révèle la grande variété et complexité de phrases qu'on peut y retrouver. L'outil développé ici, surtout dans le contexte de ce projet de mémoire limité dans le temps, ne considère qu'un nombre restreint de structures de phrases définies par l'agencement et la séquence de mots de différentes classes grammaticales.

Un algorithme de génération automatique de textes est donc mis en place à l'Étape 2. Il crée les phrases l'une après l'autre, jusqu'à ce que le document atteigne la longueur choisie par l'utilisateur. La plupart des mots sont choisis au hasard parmi les banques de données développées à l'Étape 1 en tenant compte de leurs fréquences d'apparition dans le corpus de base. Les déterminants et autres « mots-outils » sont choisis en fonction de la structure de phrase sélectionnée au hasard par l'algorithme. Les mêmes étapes sont répétées pour chaque phrase.

Il faut se rappeler du but premier de ce projet, qui est de générer des textes automatiquement lemmatisés au moment de leur création. Cet algorithme de création de textes au hasard associe donc explicitement chacun des mots du texte avec son lemme, sa « partie de discours » ou classe grammaticale, ainsi que d'autres informations pertinentes telles que le genre et le nombre pour les noms et adjectifs, les temps et les personnes pour les verbes. Ces informations peuvent ensuite être directement comparées avec les résultats fournis lors de l'évaluation d'algorithmes de lemmatisation. De telles évaluations sont effectuées à l'Étape 3 du projet. En sortie de l'Étape 2, on a donc, en plus du texte aléatoire lui-même, un tableau contenant chacun des mots du texte accompagné de ses caractéristiques (lemme, classe grammaticale, etc.), qui sert alors de comparaison avec l'information en sortie des lemmatiseurs existants évalués à l'Étape 3.

Mais bien que le projet puisse se définir en trois grandes étapes, le rapport lui-même est composé de sept chapitres, brièvement décrits plus bas :

- Chapitre 1 : Introduction à la problématique de la recherche courante
- Chapitre 2 : Portrait des connaissances actuelles en lien avec la lemmatisation et la désambiguïsation des homographes
- Chapitre 3 : Méthodologie adoptée pour la lemmatisation (Étape 1) et la génération de textes automatiquement lemmatisés (Étape 2)
- Chapitre 4 : Algorithmes et éléments de programmation, incluant des extraits de code en langage Java en lien avec les objectifs visés
- Chapitre 5 : Présentation des résultats autant pour la lemmatisation du corpus de base, incluant le traitement syntaxique des homographes (Étape 1), et la génération de textes aléatoires automatiquement lemmatisés (Étape 2)
- Chapitre 6 : Évaluation de lemmatiseurs existants (Étape 3)
- Chapitre 7 : Conclusion de ce projet de recherche avec retour sur les objectifs visés

Ce mémoire contient finalement quelques annexes apportant un complément d'information sur certains sujets précis. Ces informations, bien que faisant partie intégrante de l'ouvrage, y ont été logées afin d'alléger la lecture du texte principal.

2. PORTRAIT DES CONNAISSANCES

Le traitement automatique des langues naturelles (TALN) est un domaine en pleine évolution, où se mêlent et convergent la reconnaissance vocale, la génération automatique de textes, la traduction automatique, la recherche par mots-clés, et ainsi de suite. Ce chapitre donne un aperçu de l'état des connaissances pour une sous-classe seulement du TALN, plus précisément la lemmatisation et la désambiguïsation automatique des homographes.

2.1. Lemmatisation

La lemmatisation consiste à associer chaque mot à son lemme ou forme canonique. Ainsi, il devient possible d'associer plusieurs formes fléchies à un seul lemme commun. Mais la lemmatisation n'est pas la seule façon d'y arriver. Une autre option, appelée troncation (*stemming* en anglais), poursuit ce même but par une approche simple et directe. La troncation consiste à réduire tous les mots ayant le même radical à une forme commune en retirant la terminaison. Cette méthode simple a été comparée avec l'approche plus avancée de lemmatisation par Balakrishnan et Lloyd-Yemoh (2014) pour la langue anglaise. Bien qu'elles aient noté une meilleure performance de la lemmatisation par rapport à la troncation, la différence observée n'était pas significative. Pour des langues plus complexes que l'anglais au niveau morphologique, la performance de la troncation n'est en revanche pas aussi bonne. Perera et Witte (2005) déplorent que la troncation, appliquée à l'allemand, se retrouve à générer des formes de mots qui en fait n'existent pas dans la langue (« *overstemming* »). Malgré tout, la troncation demeure utilisée dans cette langue par des outils de base servant à la recherche d'information. Hein (1990) relève quant à elle la limite de la troncation dans le cas des mots composés dans la langue suédoise, principe commun à d'autres langues germaniques. Dans un tel cas, il est en effet difficile d'associer le bon lemme à de tels mots issus de la combinaison de deux mots initiaux. Le français regorge aussi d'exemples qui démontrent les limites de la troncation, par exemple pour lemmatiser les différentes formes du verbe « aller », provenant de radicaux bien différents (« allons », « va », « irai ») ou du verbe « être » (« es », « suis », « serai »). Mais de toute façon, même dans les cas où la troncation peut s'avérer efficace, elle ne peut informer sur la partie de discours du terme tronqué, une autre fonction recherchée par la lemmatisation.

La lemmatisation a en effet comme objectif additionnel d'associer chaque mot à sa partie de discours. On parle alors d'*étiquetage*, qui consiste à associer à chaque mot non seulement sa classe grammaticale, mais aussi d'autres paramètres tels que son genre et son nombre pour les noms communs et adjectifs, et son temps et sa personne pour les verbes, et aussi le cas (nominatif, accusatif, etc.). Tous ne s'entendent pas sur les classes grammaticales précises selon lesquelles distinguer les mots. Gendner et Adda Decker (2002) considèrent par exemple huit classes grammaticales principales (verbes, noms, adjectifs, adverbes, déterminants, pronoms, prépositions, et conjonctions) tandis que Bourdaillet et Ganascia (2005) ajoutent les classes plus spécialisées « mot-phrase », « résidu », « ponctuation » et « extra-lexical ». Le nombre de classes influe évidemment sur le nombre d'étiquettes distinctes. Vergne (1998) discute justement des avantages relatifs au nombre d'étiquettes. Il soulève que bien qu'un grand nombre d'étiquettes permette une classification plus précise donc plus utile pour usage ultérieur, le taux de succès lors de la lemmatisation est forcément plus faible que lorsque moins d'étiquettes sont imposées. Un défi de l'étiquetage est de comparer des études effectuées par différents auteurs basées sur des étiquetages différents, ce qui exige une projection des résultats d'une étude sur l'autre, un processus soit manuel ou demandant un algorithme de transfert (Vergne, 1998).

Différentes approches ont été proposées pour déterminer l'étiquette de tous les mots d'une phrase, et l'une d'entre elles est de considérer la phrase dans son entier. On analyse alors son

« arbre syntaxique », qui permet selon Yvon (2010) d'identifier les frontières des constituants de la phrase, par exemple le groupe nominal et le groupe verbal, ainsi que les relations que ces groupes entretiennent entre eux. Pinker (2015) insiste aussi sur cet intérêt de visualiser la structure des phrases grâce aux arbres syntaxiques, en illustrant l'ordre des chaînes de mots d'une phrase aux extrémités de « branches » d'un arbre inversé dont la racine est située dans le haut. Ces arbres permettent en effet de regrouper les différentes clauses et ultimement d'étudier le sens d'une phrase, en particulier dans le cas où celui-ci pourrait s'avérer ambigu.

Tout comme Yvon et Pinker, Vergne (1998) souligne aussi l'intérêt de procéder à l'analyse syntaxique de la phrase en entier, plutôt que de se limiter à une analyse plus locale autour du mot à lemmatiser, dite *contextuelle*. Pour y arriver, et donc créer l'arbre syntaxique, Vergne suggère l'utilisation d'une banque de « structures attendues » pour les phrases. Mais cette approche pose le problème de l'exhaustivité, puisqu'il est difficile de générer toutes les variations possibles dans la construction des phrases du français. Vergne suggère donc aussi une approche permettant les retours en arrière dans l'analyse de la phrase, pour arriver à des algorithmes efficaces de génération d'étiquettes morpho-syntaxiques.

Une fois ces grandes classes grammaticales et étiquettes plus précises définies, on peut en analyser la fréquence. Gendner et Adda-Decker (2002) fournissent en effet les proportions relatives des huit classes grammaticales qu'ils ont définies, en termes de fréquence dans les textes. Ils constatent que les classes les plus fréquentes sont les noms, les déterminants, les prépositions et les verbes, comportant chacune plus de 10% des mots rencontrés dans divers corpus. Ce constat est important selon eux, car c'est aussi au sein de certaines de ces classes (noms, verbes et adjectifs) que l'on retrouve le plus de variations dues aux flexions. Il est toutefois à noter que ces proportions concernent les fréquences totales dans un corpus, et non le nombre de formes distinctes, qui sont par exemple très réduites dans le cas des déterminants et pronoms, en comparaison avec les noms et les verbes. Liu et al. (2012) mentionnent toutefois que tous les chercheurs ne s'entendent pas nécessairement sur la classe grammaticale du lemme à associer à certains mots, citant le cas des adverbes. En effet, Liu et al. relèvent que certains auteurs associent certains adverbes à leurs formes adjectivales associées. En français par exemple, on pourrait associer l'adverbe « lentement » à sa forme adjectivale « lent », plutôt que de considérer l'adverbe comme étant son propre lemme.

Cependant, l'étiquetage requiert l'usage de grandes banques lexicales. Tel que le mentionne Perera et Witte (2005), deux grandes approches sont généralement proposées pour fournir cette information de base requise pour la lemmatisation. On fait soit appel à des banques exhaustives de mots fléchis, ou encore on applique des règles servant à former les formes fléchies à partir d'un radical donné. Créer de grandes banques lexicales de formes fléchies est un exercice fastidieux, tout de même exécuté au cours du temps par différentes équipes de chercheurs. Par exemple, Perennou et de Calmès (1987) ont mis au point la base de données relationnelle BDLEX (Base de Données LEXicales) dans le but de faciliter le traitement automatisé du français écrit et parlé. La version originale de cette base de données (BDLEX-0) comprenait 7000 entrées lexicales et 15,000 mot fléchis. Ferrané et al. (1992) ont d'ailleurs appliqué avec succès les versions subséquentes BDLEX-1 et -2 à un large corpus de textes (BREF) établi à partir d'articles de journaux, pour en tester les taux de couverture, donc leur étendue. Un peu plus récemment, Hug (2002) mentionne la très large banque de données FRANTEXT pour le français, qu'il utilise pour ses travaux. D'autres banques de données en français telles que GRACE, utilisée par Bourdaillet et Ganascia (2005) et la banque de syntagmes figés CLAPI utilisée par Tutin (2019), sont aussi couramment citées. Toujours est-il que Perera et Witte (2005) ont plutôt choisi la deuxième option, consistant à l'application de règles pour la création de banques de lemmes de mots fléchies. Ils y sont arrivés grâce à un outil qu'ils ont développé pour la langue allemande. Un tel outil a l'avantage de ne pas dépendre de l'exhaustivité de banques de mots existantes en plus

de facilement permettre le traitement de néologismes. Mais de façon générale, un tel outil ne peut être utilisé seul, comme ces auteurs le soulignent. Il faut aussi faire appel à un lexique de base pour complémenter le tout. Finalement, à mi-chemin entre les banques de données et les outils de formation de mots, on retrouve le classique « Bescherelle » (Bescherelle, 2012). Brunet (2017), le présentant à la fois comme un outil de conjugaison et de *dérivation*, insiste sur le fait qu'il permet de générer un relevé exhaustif des formes possibles pour un même verbe, dans le contexte de la lemmatisation. Brunet souligne d'ailleurs que les créateurs de la banque FRANTEXT ont fait ce choix de mettre en mémoire toutes les formes verbales possibles.

Au-delà des formes verbales issues par exemple du « Bescherelle », il est très pratique d'aussi identifier ce qu'on appelle des syntagmes figés. On définit un syntagme comme un ensemble de mots formant une seule unité fonctionnelle. Dans un cas général, l'association entre les mots composant le syntagme est dite « libre », car occasionnelle et spécifique au contexte dans lequel on le retrouve. Mais lorsque des mots sont régulièrement associés entre eux, d'une façon telle qu'ils en perdent leur individualité et leur fonction individuelle, on parle de syntagme *figé*. On pense par exemple aux formes « pomme de terre », « livre de poche », ou « prendre la poudre d'escampette ». Gross (2004) insiste d'ailleurs sur l'importance et l'intérêt d'identifier ces formes lorsque présentes dans un texte pour en faire l'analyse statistique. Des algorithmes doivent être mis au point pour identifier celles-ci, et en particulier d'identifier la version la plus longue possible d'un syntagme figé existant. Pour y arriver, on doit puiser dans des banques existantes ou *dictionnaires* de tels syntagmes, tel que le CLAPI cité plus haut. La présence de syntagmes n'est évidemment pas unique au français. Hein (1990) exprime par exemple l'importance des mots composés en suédois de type « nom-nom » qu'on peut aussi associer à des syntagmes. L'existence des nombreux mots composés illustre selon elle les restrictions existantes dans la combinaison des mots. Aussi, toujours selon Gross (2004), l'existence de syntagmes figés nous porte à nous questionner sur ce qu'on entend par un « mot », puisque ces syntagmes sont dans la plupart des cas utilisés comme éléments unitaires dans une phrase, qui pourraient donc être remplacés par un seul autre « mot ». Compter le nombre de mots dans un texte n'est donc pas une tâche triviale selon lui, se limitant à y identifier les « espaces blancs ». Il s'agit au contraire d'une tâche contenant son lot de complexité.

Les banques de mots et syntagmes cités plus haut se rapportent à la langue générale, puisqu'ils sont généralement issus de textes génériques. Mais Yvon (2010) déplore que ces banques de mots générales ne fournissent pas toujours un traitement complet pour des textes portant sur des domaines de connaissances précis. Ces banques ne sont donc pas indépendantes du domaine. Liu et al. (2012) insiste aussi sur l'importance de développer des banques de mots de domaines précis pour lemmatiser les textes s'y rapportant. Ils ont en effet développé un outil de lemmatisation spécialisé pour les textes du domaine biomédical, sur la base d'un lexique inspiré de ce domaine. Ils ont démontré que la performance de leur outil dépasse celle des lemmatiseurs basés sur des banques de mots générales. Grouin (2022) insiste aussi sur le besoin de bâtir des lexiques pour les langues de spécialité, citant en exemple les domaines juridique et médical. Mais il ajoute qu'il est aussi approprié de considérer des banques de mots spécialisées pour les registres de langue (académique, familier, soutenu) ou en lien avec des usages particuliers (courriers électroniques, réseaux sociaux, messages textuels par téléphone). Ces différents registres peuvent en effet incorporer des mots ne faisant pas partie des lexiques généraux. L'Homme (2008) résume le tout en affirmant qu'il a été démontré que « l'analyse de ces productions langagières au moyen d'outils de traitement automatique des langues est de meilleure qualité si les ressources développées sont représentatives des usages qu'elles doivent traiter ».

Une autre difficulté rencontrée lors de la lemmatisation d'un texte, en dépit de larges de banques de mots spécialisées ou non, est l'identification des noms propres. Cette difficulté tient de

l'impossibilité d'en dresser un inventaire exhaustif, vu leur nombre, mais surtout considérant le fait qu'il s'en crée constamment de nouveaux, à mesure que les humains baptisent des gens et des lieux. Hein (1990) par exemple, note qu'environ 6% des mots non identifiés d'un texte, donc ceux ne pouvant pas être lemmatisés, sont des noms propres, au sein des textes suédois qu'elle a analysés. Contamine (1975), dans son analyse de textes latins, réfère à ces mots non facilement classifiables dont les noms propres et les mots avec orthographe variable, comme des « rebuts » des algorithmes de lemmatisation. Elle souligne que ces rebuts doivent être traités à la main, mais qu'heureusement, ces formes non-classifiées diminuent avec l'analyse cumulée de différents corpus. En effet, cette analyse manuelle alimente en temps réel les banques de mots sur lesquelles sont basées les analyses. Mais Vergne (1998) propose quant à lui la mise en place d'algorithmes visant à déterminer la classe grammaticale de mots non reconnus, sur la base de l'analyse syntaxique. De tels algorithmes visent justement à minimiser le recours à une analyse manuelle. Il réfère à ces algorithmes comme étant des « *guessers* », terme anglais couramment utilisé en français dans ce contexte. De tels « *guessers* » peuvent donc servir soit à complètement automatiser l'identification de mots non autrement classifiés, ou à tout le moins à en suggérer la classe grammaticale aux chercheurs.

On voit donc que la lemmatisation d'un texte ne se fait pas sans heurt, vu tous les défis se présentant, incluant par exemple la présence potentielle de fautes d'orthographe et de grammaire dans les textes, tel que souligné par Grouin (2022). Brunet (2017) insiste ainsi sur le fait que la lemmatisation demeure une opération couteuse, et qu'il faut en évaluer la pertinence selon le besoin du travail de recherche. Mais heureusement, comme il s'en réjouit, de nombreux outils (certains décrits à la section suivante) sont maintenant disponibles pour en faciliter l'exécution.

2.1.1. Outils de lemmatisation actuels

Comme on ne doit pas continuellement réinventer la roue, de nombreux outils de lemmatisation développés au cours des années ont été diffusés et employés par plusieurs chercheurs. Certains sont spécifiques à une langue, comme le LPS (*Lexicon-Oriented Parser*) cité par Hein (1990) pour le suédois. Mais d'autres outils ont démontré leur grande versatilité, ayant été adaptés pour usage dans plusieurs langues. En effet, afin de bien classifier les différents mots d'une phrase selon leur partie de discours ou classe grammaticale, des chercheurs spécialisés dans plusieurs langues font appel au puissant outil TreeTagger (Schmid, 1995). Perera et Witte (2005) s'en sont servi entre autres pour maximiser le nombre de noms lemmatisés, en particulier dans le cas dit « nominatif » en allemand, lorsque le nom est le sujet de la phrase. Ils ont combiné l'analyse fournie par TreeTagger avec celle fournie par l'algorithme qu'ils ont eux-mêmes développé. Grouin (2022), comme plusieurs autres chercheurs francophones, utilise aussi l'outil TreeTagger pour l'étiquetage de mots en français. Grouin suggère aussi un autre outil plus récent (spaCy), bien que moins performant que le TreeTagger dans un contexte général, qui s'est montré plus efficace dans son analyse du français *inclusif*, comme quoi chaque outil a ses points forts et points faibles. Pour le français toujours, un des lemmatiseurs les plus utilisés est l'outil « Cordial ». Brunet (2017) mentionne en effet que cet outil est intégré à la version française de Microsoft Word. Bourdaillet et Ganascia (2005) jugent aussi que Cordial est présentement l'outil le plus puissant pour lemmatiser les textes français.

Mais en plus de se soucier de la langue pour laquelle la lemmatisation est nécessaire, on doit aussi se soucier du domaine de connaissances, tel que mentionné à la section précédente. Liu et al. (2012) ont en effet comparé la performance de huit outils de lemmatisation différents, dans leur recherche portant sur les textes biomédicaux en anglais. On en conclut, là encore, que certains outils performent mieux que d'autres dépendamment du contexte. Il n'est donc pas surprenant que pour leurs propres besoins précis, des chercheurs en viennent à développer leurs propres outils de lemmatisation, comme Liu et al. l'ont fait pour le domaine biomédical.

L'identification de la partie du discours (classe grammaticale), une tâche pour laquelle TreeTagger, Cordial et les autres outils cités plus haut excellent, facilite grandement la détermination subséquente du lemme. C'est particulièrement le cas pour les homographes, sujet de la section suivante.

2.2. Désambiguïsation des homographes

Tel qu'on l'a constaté à la section précédente, la lemmatisation s'accompagne de plusieurs défis. Mais probablement le plus grand de ces défis se rapporte à la *désambiguïsation* des homographes. Les homographes sont des mots dont la graphie est la même, mais dont le sens et le plus souvent la classe grammaticale diffèrent. Il est en effet impossible de désambiguïser ou distinguer ces homographes sans considérer le contexte dans lequel ils sont utilisés. En effet, Nubel et Pease (2002), dans le contexte de l'allemand, insistent sur le fait que la lemmatisation au niveau du mot seul, laisse place à beaucoup d'ambiguïtés, ce qui complique les tâches de traduction. On retrouve des homographes dans plusieurs langues, anciennes comme contemporaines. Brunet (2017) mentionne que ceux-ci sont plus fréquents en latin qu'en français, et qu'ils engendrent ce qu'il appelle une « pollution diffuse » quand vient le temps d'effectuer la lemmatisation. Brunet insiste aussi sur la difficulté de lemmatisation et de désambiguïsation des textes du Moyen-Âge, considérant que l'orthographe à l'époque était « flottante » et l'accentuation « fantaisiste ». Contamine (1975) cite en exemple la difficulté de désambiguïser en latin les cas opposant les formes adjectivales et les participes passés. Hein (1990) insiste aussi sur le défi représenté par la résolution des homographes, dans son cas en suédois, dont les types et fréquences sont très variés. Les homographes sont en fait si nombreux, dans les langues anciennes comme dans les langues modernes, que les désambiguïser manuellement est devenue une tâche « inhumaine » pour reprendre les mots de Brunet, pour des textes de grande taille.

Hug (2002), reconnaissant lui aussi cette tâche colossale, insiste sur la pertinence de la désambiguïsation selon le but recherché. En effet, certaines activités, incluant l'analyse sémantique, ne requièrent pas nécessairement une désambiguïsation parfaite. En effet, qu'un homographe soit issu d'une forme nominale ou verbale peut dans bien des cas ne pas affecter la sémantique. Comme le mentionne Hug, le sens associé aux homographes de type « nom-verbe » est souvent similaire. On peut citer en exemple, le nom « demande » et la forme verbale « demande ». Mais ce n'est pas toujours le cas. On pense au verbe « laisse » qui n'a pas le même sens que le nom « laisse », comme le cite Hug. Il incombe donc aux utilisateurs ou chercheurs de juger de la pertinence de la désambiguïsation pour leur propre travail, compte tenu de l'effort impliqué.

De la même façon qu'on l'a fait pour faciliter la lemmatisation en général, deux approches ont été adoptées pour la désambiguïsation des homographes, tel que mentionné par Contamine (1975). Ou bien on développe des « dictionnaires d'homographes » qu'on accroît à mesure de l'analyse de nouveaux textes, ou bien on développe un « code grammatical » qui fournit les règles pour la flexion des homographes. Mais surtout, il faut considérer le fait que si la plupart des homographes sont issus de classes différentes, par exemple « nom » et « verbe », certains autres appartiennent à la même classe grammaticale. Pour le suédois, Hein (1990) a estimé que 15% des homographes sont issus de la même classe. En français, on pourrait parler des mots comme « fils », soit un nom singulier comme enfant de sexe masculin ou un nom pluriel représentant un bout de tissu, ou encore la forme verbale « suis », venant soit du verbe « être » ou du verbe « suivre ». Il va de soi que la stratégie à adopter pour la désambiguïsation d'homographes issus de la même classe grammaticale diffère de celle à privilégier quand les classes diffèrent.

Pour les classes grammaticales différentes, on fait appel à la syntaxe de la phrase pour faciliter la désambiguïsation. On peut en effet développer des arbres syntaxiques décrits à la section

précédente, pour déterminer la classe grammaticale d'un mot, ce qui permet ensuite dans bien des cas d'en déterminer la bonne forme homographique. Pour effectuer la désambiguïsation de façon automatique, grâce à ces arbres ou non, il faut mettre au point un certain nombre de règles précises. Vergne (1998) classe ces règles en deux catégories : les déductions positives et les déductions négatives. Une déduction positive consiste en un test qui, lorsque vérifié, confirme qu'un mot appartient à une classe en particulier, tandis qu'une déduction négative concerne l'impossibilité qu'un mot, dans un contexte particulier, puisse appartenir à une certaine classe grammaticale. Hug (2002), s'attaquant en particulier aux homographes de type « nom-verbe » pour la langue française, a énoncé certaines règles pertinentes. Certaines de ces règles sont basées sur le nombre et le genre des mots environnants, ou la présence de mots appartenant de façon certaine à certaines classes dans l'environnement de l'homographe étudié.

Mais en plus de ces règles syntaxiques, d'autres indices peuvent être utilisés, en lien avec les statistiques de la langue. Par exemple, Vergne (1998) mentionne qu'en observant la proportion de déterminants et de pronoms pour certains homographes français en particulier dans de larges corpus, on peut éviter l'analyse syntaxique et supposer que chaque homographe rencontré appartient à la classe la plus probable pour ce mot. Une telle analyse simpliste implique forcément un certain nombre d'erreurs. Mais autant Nubel et Pease (2014) que Hug (2002) minimisent l'impact global de ces erreurs de désambiguïsation pour un texte en entier, reconnaissant qu'une certaine proportion d'erreurs peut être considérée acceptable, dépendamment du contexte. Hug déplore en revanche que certains lemmatiseurs soutiennent que certains homographes ont été désambiguïsés de façon « certaine », alors qu'en fait, on retrouve encore parmi ceux-ci bon nombre d'erreurs. Il faut donc se méfier de ce que certains outils de lemmatisation fournissent comme information en sortie, une mise en garde partagée par Brunet (2017), qui a analysé l'information fournie en sortie par un de ces outils.

Vergne (1998) et Hug (2002) suggèrent aussi l'utilisation de banques de syntagmes récursifs pour faciliter la tâche de désambiguïsation, car en présence de tels syntagmes ou locutions, il est facile de déterminer la classe grammaticale de chaque mot les constituant. En effet, plusieurs syntagmes impliquent l'homographe « de », qui peut être soit un déterminant ou une préposition. La présence du mot « de » dans un syntagme permet au moins dans ces cas précis de bien le désambiguïser. Les banques de syntagmes citées à la section précédente peuvent donc s'avérer très utiles pour traiter les homographes.

Mais, comme le précise Hein (1990), les homographes les plus difficiles à désambiguïser sont ceux issus de la même classe grammaticale, et cette difficulté du suédois se retrouve aussi pour le français. En effet, dans de tels cas, il n'est généralement pas possible de faire appel à des règles de grammaire pour désambiguïser de tels homographes, ceux-ci étant « indifférents » à l'analyse syntaxique. Toutefois, comme le mentionne encore Hein, certains de ces homographes de même classe sont des noms pouvant être désambiguïsés sur la base de leur genre. C'est aussi le cas pour certains mots français comme « mousse », qu'on retrouve soit au masculin ou au féminin, avec deux sens différents. Dans le cas de la forme verbale « suis », il est parfois possible de la désambiguïser, dans les cas où on peut prouver que le verbe est conjugué à la deuxième personne du singulier, puisque la forme « suis » ne se retrouve pour le verbe « être » qu'à la première personne du singulier. Il s'agit donc dans un tel cas du verbe « suivre ». En suédois comme en français, le nombre peut aussi être utilisé pour désambiguïser, comme pour l'exemple « fils » cité plus haut. Mais dans un cas plus général, c'est souvent uniquement par la sémantique que la désambiguïsation d'homographes issus de la même classe grammaticale peut être effectuée. Il va sans dire qu'une telle analyse peut s'avérer très complexe.

2.2.1. Apprentissage machine pour la désambiguïsation

Les récents pas de géant en intelligence artificielle représentent une bonne nouvelle pour le traitement automatique des langues naturelles en général, mais aussi pour le cas particulier de la désambiguïsation des homographes. En effet, vu la tâche colossale de la lemmatisation de larges textes et la proportion importante d'homographes parmi ceux-ci, l'apprentissage machine, qui fonctionne à son mieux avec d'imposantes quantités de donnée, est une solution bien appropriée.

Mais les chercheurs en linguistique informatique n'ont pas attendu cette récente embellie de l'intelligence artificielle pour incorporer des outils d'apprentissage machine pour la désambiguïsation des homographes. En effet, des études datant de plusieurs années faisaient déjà état de l'utilisation d'outils informatiques avancés pour la résolution d'homographes. Par exemple, pour l'étiquetage morpho-syntaxique, Perera et Witte (2005) ont fait appel au Modèle de Markov Caché (« *Hidden Markov Model* », HMM en anglais), un outil statistique cherchant à déterminer les séquences les plus probables. Dans le cas qui nous concerne, cela peut être des séquences de classes grammaticales dans les phrases. Le HMM sert en effet, en plus de l'étiquetage, à la traduction automatique et à la reconstruction de textes bruités. Perera et Witte (2005) l'ont appliqué aux cas nominatif, accusatif, datif et génitif des phrases allemandes pour former une multitude d'étiquettes de structures de phrases. À la même époque, Bourdaillet et Ganascia (2005) ont quant à eux appliqué le HMM à la langue française, en définissant leur jeu de 320 étiquettes morpho-syntaxiques comme autant d'états du système.

Ingason et al. (2008) ont de leur côté introduit une approche différente d'apprentissage machine appelée « *Hierarchy of Linguistic Identities* – HOLI ». Leur objectif était de s'éloigner des lemmatiseurs généraux applicables à plusieurs langues, pour bâtir un outil spécialisé pour l'islandais, en l'arrimant à des banques de données spécifiques à cette langue riche en formes fléchies. Un des défis auquel ces chercheurs ont dû faire face est la présence abondante de mots composés, tâche pour laquelle leur outil HOLI s'est avéré efficace, vu la grande quantité de données disponibles.

Plus récemment, Kutuzov et Kuzmenko (2019) ont introduit des éléments d'apprentissage profond pour la lemmatisation, plus particulièrement dans leur cas pour la désambiguïsation de sens des mots. Ils ont découvert que la réduction des mots à leurs lemmes contribuait fortement à leur mise en contexte, observation contraire à ce que d'autres chercheurs croyaient jusque-là, à savoir que la lemmatisation n'était pas critique pour ce genre d'opérations, du moins dans le cas de la langue russe. Kutuzov et Kuzmenko (2019) concluent donc que la lemmatisation et donc la préparation de textes, supportée par l'apprentissage profond, demeure un outil très utile.

Encore plus récemment, Iqbal et Qureshi (2022) ont effectué une recherche sur les outils existants en apprentissage profond appliqués au traitement automatique des langues naturelles et en particulier à la génération de textes. Tout comme Kutuzov et Kuzmenko (2019) cités plus haut, Iqbal et Qureshi considèrent aussi que les modèles d'apprentissage machine fonctionnent à leur meilleur à partir de données bien étiquetées. Mais ils reconnaissent évidemment que les données de base disponibles sont typiquement non étiquetées, il importe de mettre au point des méthodes d'entrainement *non supervisées* qui permettent de gagner énormément de temps.

En conclusion, pour rendre la tâche de désambiguïsation des homographes moins « couteuse », pour reprendre les mots de Brunet (2017), l'appel à des algorithmes d'apprentissage machine puissants, idéalement non supervisés, question de minimiser l'effort manuel, est de plus en plus l'approche à suivre.

Aucune autre référence pertinente plus récente n'a pu être identifiée en lien avec ce travail de recherche.

3. MÉTHODOLOGIE

Au cours de ce chapitre, on décrit l'approche et la méthodologie générales adoptées pour les différentes étapes de ce projet, sans égard toutefois à comment ces méthodes seront mises en application dans le contexte d'un outil informatique. Ce chapitre offre donc une bonne vue d'ensemble des objectifs et des défis inhérents à ce projet, sans encore trop s'attarder aux détails, qui seront quant à eux discutés au Chapitre 4, qui se penchera plus en profondeur sur les algorithmes.

3.1. Étape 1 : Outil de lemmatisation de base

La première partie de ce mémoire (l'Étape 1) consiste à créer un outil de lemmatisation de textes en français. Il peut paraître contradictoire ou paradoxal de concevoir un outil de lemmatisation dans le but ultime d'évaluer d'autres outils de lemmatisation. Mais il faut considérer que l'outil de lemmatisation créé dans le cadre de ce projet n'a pas besoin d'être particulièrement efficace, car l'objectif premier de cet outil est d'abord de générer une grande liste de lemmes (verbes, noms communs, adjectifs et adverbes) représentatifs du corpus de référence choisi. En effet, il n'est pas strictement nécessaire que les mots extraits soient exactement ceux correspondant au texte. Autrement dit, la précision et l'efficacité de cet algorithme de lemmatisation n'est pas critique. Il faut toutefois s'assurer que les lemmes extraits sont bien associés aux bonnes classes de mots : un nom est bien identifié comme un nom, un verbe comme un verbe, etc.

Un certain travail manuel est nécessaire en amont afin de bâtir des banques de données de différentes classes grammaticales. Ces banques de données sont utilisées dans le but de reconnaître chaque mot rencontré dans le corpus, et d'en déterminer la classe grammaticale ainsi que certains autres paramètres s'y rapportant. Mais cet effort n'est requis qu'une seule fois pour un corpus de référence donné. À mesure que de nouveaux corpus seront intégrés à l'analyse ou que le corpus sera élargi, les besoins de mise à jour des banques de données iront en diminuant, car y figureront de moins en moins de *nouveaux* mots non encore inclus dans les banques de données précédentes.

Bien sûr, un algorithme de lemmatisation existant aurait pu jouer le rôle d'extraire les lemmes représentatifs d'un corpus donné. Mais la création ici d'un outil dédié a permis d'y inclure des fonctionnalités particulières nécessaires pour la deuxième étape du mémoire. En particulier, l'outil de lemmatisation créé ici permet d'identifier des cooccurrences pour tous les verbes, noms communs, adjectifs et adverbes rencontrés dans le corpus. Ces cooccurrences, qu'on définit pour ce travail comme autres mots retrouvés dans l'environnement immédiat d'un mot étudié, sont ensuite utilisées pour bâtir à l'Étape 2 des phrases comportant un *minimum* de sens.

Aussi, l'outil de lemmatisation calcule les fréquences d'apparition des différents lemmes ainsi que les fréquences des temps et des personnes pour les verbes. Ces observations servent à la sélection de mots à l'Étape 2, pour que les textes générés au hasard soient représentatifs du corpus de référence non seulement pour le lexique, mais aussi au niveau des temps de verbe et personnes. Ainsi, un mot se retrouvant fréquemment dans le corpus de base se retrouvera aussi fréquemment dans le texte généré aléatoirement. Réciproquement, un mot n'apparaissant que rarement dans le corpus n'apparaitra aussi que rarement dans le texte généré au hasard. De plus, l'outil de lemmatisation créé calcule la longueur moyenne des phrases, ce qui est utile pour comparer avec les phrases créées aléatoirement à l'Étape 2.

Finalement, la création de ce nouvel outil de lemmatisation, bien qu'imparfait, apporte un élément d'apprentissage facilitant la compréhension de la problématique globale et une meilleure appréciation des défis y étant reliés. Une telle appréciation permet en retour une évaluation plus

pertinente et plus rigoureuse d'outils de lemmatisation existants, puisqu'on peut alors soumettre ceux-ci à des textes incorporant expressément des mots et expressions de la langue française reconnus à l'Étape 1 du mémoire comme posant difficultés. Le détail de l'implémentation de cet outil de lemmatisation en langage informatique Java est discuté au Chapitre 4.

3.1.1. Classes grammaticales et étiquetage

La lemmatisation exige en premier lieu qu'on puisse assigner à chaque mot rencontré dans un texte une classe grammaticale. Pour les besoins de ce projet, le document « Les classes de mots (mémento pour l'enseignant) » (Gomila & Fonvielle, 2018) a servi de référence pour distinguer neuf classes de mots. Ces classes sont listées au Tableau 3.1, accompagnées d'un code auquel on associe chacune de ces classes tout au long de ce document. La lemmatisation d'un mot grâce à l'outil développé ici fournit donc non seulement le lemme de ce mot (voir Section 3.1.2), mais aussi la classe grammaticale associée à ce lemme.

Il est à noter que le regroupement de tous les mots de la langue française en neuf classes générales n'est pas la seule approche possible et ne fait pas l'unanimité. Par exemple, Bourdaillet et Ganascia (2005) font référence à un large corpus de langue française dont les mots ont été regroupés en douze classes générales, plutôt que les neuf suggérées plus haut. On retrouve par exemple dans le regroupement mentionné par ces auteurs les classes « mot-phrase », « résidu », et « ponctuation », absentes du Tableau 3.1. Mais on y retrouve surtout les mêmes grandes classes principales comme les « noms », « verbes », « adjectifs », « pronoms », etc. qui composent la vaste majorité des mots du lexique français. La sélection ici de ces neuf classes, bien que relativement arbitraire, a répondu aux besoins essentiels de désambiguïsation des homographes, but ultime de ce travail.

Les classes du Tableau 3.1 peuvent ensuite être affinées, pour aboutir à un jeu d'étiquettes plus précis, dont on s'est servi pour le projet actuel. Le Tableau 3.2 résume cet affinement additionnel. À chacune de ces sous-classes est associé un code numérique (colonne de droite du tableau), déterminé arbitrairement, utilisé pour la programmation des algorithmes informatiques. Finalement, pour certaines classes grammaticales, un étiquetage encore plus précis est fourni : l'étiquetage morpho-syntaxique. Celui-ci spécifie un certain nombre de paramètres pour chaque mot. Le Tableau 3.3 fournit la liste de ces paramètres pour chacune des classes.

Chaque verbe est donc ainsi étiqueté selon le temps (présent, imparfait, futur simple, etc.) et la personne (1ère, 2e et 3e personnes du singulier et du pluriel). Dans le cas du participe passé, alors que la personne n'est pas pertinente, ce sont le genre et le nombre qui sont fournis. Le Tableau 3.4 fournit la liste des temps de verbe accompagnés d'un code utilisé pour ce projet. Il est à noter que les dix derniers temps (à partir de l'infinitif passé) sont des temps composés, ne servant donc pas à étiqueter chaque mot-verbe individuellement, mais plutôt un groupe de mots-verbe. Ces temps composés sont utilisés à l'Étape 2 pour la génération de phrases aléatoires. Le Tableau 3.5 fournit quant à lui la liste de codes pour les personnes, tandis que le Tableau 3.6 décrit les codes utilisés pour le genre et le nombre.

Chaque adjectif ou nom est associé à un genre et à un nombre, utilisant les mêmes codes que ceux listés au Tableau 3.6. Pour les adjectifs, un paramètre additionnel est nécessaire, pour déterminer si un adjectif peut être « antéposé », c'est-à-dire s'il peut se retrouver devant le nom qu'il modifie (« le *grand* garçon ») plutôt qu'après le nom (postposé), comme c'est majoritairement le cas en français (« le camion *rouge* »).

Les déterminants, devant pour la plupart s'accorder avec le nom qu'ils modifient, doivent aussi majoritairement être étiquetés en fonction de leur genre et nombre. Mais certains déterminants sont aussi caractérisés par la personne à laquelle ils font référence (1ère, 2e et 3e personnes du singulier et du pluriel), tel que listé au Tableau 3.5. En effet, les déterminants possessifs doivent

faire référence à la personne ou à l'objet qui possède le nom référencé. Par exemple, les déterminants « mon », « ton », « son », « notre », « votre » et « leur » peuvent tous être utilisés devant un nom au masculin singulier, mais font référence aux six différentes personnes du Tableau 3.5, respectivement. Il est à noter les déterminants numéraux sont invariables et donc pas associés à un genre ou un nombre (« Les *trois* petits cochons (…) »).

Pour les pronoms, on se doit aussi de préciser, en plus du genre, du nombre et de la personne, le *cas*. Le cas sert à déterminer la fonction que le pronom occupe dans la phrase. Pour un pronom-sujet, le cas est « nominatif » (« *je* mange une pomme »). Quand un pronom joue le rôle de complément d'objet direct, le cas est « accusatif » (« je *la* mange »). Pour un complément d'objet indirect, c'est le « datif » (« je *lui* donne mon livre »). Le Tableau 3.7 résume ces trois cas et fournit les codes numériques arbitraires employés pour le développement des algorithmes.

Cet étiquetage précis des mots du lexique, utilisant les codes listés aux Tableaux 3.1 à 3.7, est critique pour l'Étape 2 de ce projet consistant à générer au hasard des phrases complètes automatiquement lemmatisées, tout en respectant la structure et la syntaxe de la langue française.

En considérant toutes les classes grammaticales, toutes leurs sous-classes ainsi que les différents paramètres relatifs à chaque classe, on génère ici un total de 176 étiquettes différentes possibles pour tous les mots analysés dans ce projet. À titre de comparaison, Bourdaillet et Ganascia (2005) faisaient état de 320 étiquettes distinctes pour l'analyse de leur corpus sous étude, un nombre bien plus élevé. Il est toutefois bon de noter, comme le précisent Vergne et Giguet (1998), « qu'un petit nombre d'étiquettes rend la décision plus facile (la probabilité est plus grande de tomber par hasard sur la bonne étiquette) ». Toujours est-il que la réduction du nombre d'étiquettes comporte ses limites, car, toujours selon ces auteurs, les déductions contextuelles seraient plus difficiles « par manque de finesse et de régularité dans la description des textes ». Cela étant dit, les 176 étiquettes possibles pour le projet présent sont amplement suffisantes pour atteindre notre objectif, qui ne vise pas, il faut le rappeler, une analyse syntaxique parfaite du corpus de référence.

Tableau 3.1 : Les classes grammaticales

Classe grammaticale	Description	Code
Verbe	Il exprime une action ou un état et varie en fonction du temps, de l'aspect, de la personne, du nombre, du mode et du cas. Le verbe est le centre de la phrase verbale; il sélectionne le sujet et les compléments.	1
Adjectif	Il précise le sens du nom auquel il se rapporte. Il est généralement variable en genre et en nombre. On peut généralement l'omettre dans une phrase (effaçable), sauf quand il est attribut (« il est gentil »).	2
Nom	*Nom commun* : il désigne des êtres, des choses, des idées. Il possède généralement un genre et varie la plupart du temps en nombre. Le nom commun constitue l'élément central du groupe nominal et est généralement précédé d'un déterminant. *Nom propre* : Il commence toujours par une majuscule et désigne toujours le même et unique individu, la même et unique chose. Il est généralement invariable en nombre. Il peut être précédé d'un déterminant défini (« le Canada ») mais se retrouve plus souvent sans déterminant.	3
Adverbe	Il indique des variations d'intensité, marque une caractérisation ou une quantification ou encore précise un élément du cadre spatio-temporel de la phrase. L'adverbe est invariable et est généralement mobile, effaçable et dépend d'un autre élément de la phrase qu'il modifie.	4
Déterminant	Il détermine le nom en communiquant des informations sur le genre et le nombre. Il actualise le nom en lui apportant une valeur référentielle : renvoie à un objet précis ou déjà mentionné. Le déterminant s'accorde en genre et en nombre avec le nom. Il précède le nom mais un adjectif peut s'insérer entre le déterminant et le nom. Le déterminant est obligatoire et ne se déplace pas.	5
Pronom	Il remplace généralement un groupe nominal ; il est obligatoire et ne peut être déplacé. Le pronom est variable en genre, nombre et personne.	6
Préposition	Elle établit des relations de sens entre les éléments qu'elle relie : sens spatial, instrumental, ou causal. La préposition est invariable et obligatoire. Elle se place devant le groupe nominal pour former une unité syntaxique et ne peut en être séparée sauf exception.	7
Conjonction	Elle permet de réunir des constituants et est invariable.	8
Interjection	Elle peut exprimer une sensation, un sentiment, une émotion ou bien un appel ou un ordre. L'interjection est invariable sauf exceptions et est souvent suivie, à l'écrit, d'un point d'exclamation.	9

Tableau 3.2 : Affinement des grandes classes de mots

Classe	Sous-classe	Code
	Non classifié	0
Verbe	Verbe	1
Adjectif	Adjectif	2
Nom	Nom commun	31
	Nom propre	32
Adverbe	Adverbe (non classifié)	4
	Adverbe d'intensité	41
	Adverbe de caractérisation	42
	Adverbe de quantification	43
	Adverbe spatio-temporel	44
Déterminant	Article	51
	Démonstratif	52
	Possessif	53
	Numéral	54
	Indéfini	55
	Relatif	56
	Interrogatif/exclamatif	57
Pronom	Pronom personnel	61
	Pronom démonstratif	62
	Pronom possessif	63
	Pronom indéfini	64
	Pronom relatif	65
Préposition	Préposition	7
Conjonction	Conjonction	8
Interjection	Interjection	9

Tableau 3.3 : Paramètres additionnels pour l'étiquetage morpho-syntaxique

Classe	Paramètre 1	Paramètre 2	Paramètre 3	Paramètre 4
Verbe	Temps	Personne		
Adjectif	Genre	Nombre	Antéposition	
Nom	Genre	Nombre		
Déterminant	Genre	Nombre	Personne	
Pronom	Genre	Nombre	Personne	Cas

Tableau 3.4 : Liste des temps de verbe

Temps	Code	Temps	Code	Temps	Code
Infinitif	0	Impératif	7	Passé antérieur	14
Présent	1	Conditionnel	8	Futur antérieur	15
Imparfait	2	Participe présent	9	Subjonctif passé	16
Passé simple	3	Participe passé	10	Subjonctif plus que parfait	17
Futur simple	4	Infinitif passé	11	Conditionnel passé	19
Subjonctif présent	5	Passé composé	12	Participe présent-passé	20
Subjonctif imparfait	6	Plus que parfait	13		

Tableau 3.5 : Liste des personnes

Singulier		Pluriel	
Personne	**Code**	**Personne**	**Code**
1ère du singulier	1	1ère du pluriel	4
2e du singulier	2	2e du pluriel	5
3e du singulier	3	3e du pluriel	6

Tableau 3.6 : Genre et nombre

Genre	Code	Nombre	Code
Masculin	0	Singulier	0
Féminin	1	Pluriel	1

Tableau 3.7 : Liste des cas

Cas	Fonction	Code
Nominatif	Sujet	0
Accusatif	Complément d'objet direct	1
Datif	Complément d'objet indirect	2

3.1.2. Concept de lemme

La lemmatisation d'un texte consiste à réduire chaque mot contenu dans celui-ci à sa forme de base ou canonique. Cette opération revient à grouper tous les verbes conjugués sous leur forme infinitive, tous les adjectifs sous leur forme « masculin singulier », et tous les noms communs sous leur forme au singulier. C'est d'ailleurs cette forme de base, appelée « lemme », qu'on retrouve dans les entrées de dictionnaires et non les formes dites fléchies. Le dictionnaire en ligne « Usito » (Usito, 2024) définit ainsi le lemme, dans le contexte de la linguistique, comme étant la forme *canonique* d'un mot variable.

Tous ne s'entendent pas toujours sur l'application de cette définition. En effet, Liu et al. (2012) mentionnent que certains outils de lemmatisation, lors de la lemmatisation d'adverbes, les

ramènent à leur adjectif associé. Bien que les travaux de Liu et al. portent sur la langue allemande, ce problème se pose aussi en français, où par exemple l'adverbe « lentement » pourrait être réduit à l'adjectif « lent » lors de la lemmatisation. Cependant, tout au long de ce projet, *chaque mot se verra associé à un lemme appartenant à sa propre classe grammaticale*. Ainsi, l'adverbe lentement ne sera pas associé à un lemme autre que lui-même dans ce travail.

Le Tableau 3.8 comporte quelques exemples de mots dits « fléchis » et de leurs lemmes associés. Tel que décrit plus haut, on inclut dans ce tableau un exemple illustrant le fait que les adverbes sont conservés tel quel, et ne sont donc jamais ici réduits à une forme adjectivale.

Il est à noter, tel que mentionné par Grouin (2022), que la référence à la forme d'un nom ou d'un adjectif au masculin singulier pour définir un lemme suscite la controverse. En effet, le choix du masculin est somme toute arbitraire, et va à contresens de la promotion de l'usage d'un français inclusif. Toujours est-il que pour ce projet, la forme au masculin singulier demeurera la forme de base du lemme, par souci de simplicité.

La description détaillée des algorithmes servant à la création des banques de données de mots et lemmes suivant l'étiquetage apparaissant aux Tableaux 3.1 à 3.7 est fournie au Chapitre 4.

3.1.3. Génération des banques de données

Idéalement, le lexique entier de la langue française devrait être disponible pour ce projet, dans un format se portant bien à l'analyse informatique. Au lexique de base correspondant aux mots de différentes classes grammaticales se retrouvant dans le dictionnaire, devraient s'ajouter toutes les formes fléchies possibles des noms, adjectifs et verbes. Par exemple, les pluriels « tulipes » et « grands » devraient se retrouver dans les banques de données, au même titre que leurs singuliers respectifs « tulipe » et « grand ». Aussi, les féminins « infirmière » et « bleue » devraient être listés, tout comme leurs équivalents masculins « infirmier » et « bleu ». De la même façon, toutes les formes verbales d'un verbe devraient être contenues dans les banques de données. On peut citer en exemple toutes les formes conjuguées possibles du verbe « faire » (« fais », « fait », « faisons », « ferons », etc.). Avec de telles banques de données exhaustives, un outil de lemmatisation peut associer tout mot apparaissant dans un corpus quelconque à son lemme et ultimement lui associer un étiquetage précis, c'est-à-dire une classe grammaticale, un genre, nombre, temps, et une personne, le cas échéant.

Hélas, bien que de nombreux chercheurs aient mis au point de telles banques de données pour leurs propres recherches, celles-ci ne sont pas facilement disponibles, et pas nécessairement non plus dans un format se prêtant au besoin actuel. Pour le projet présent, des banques de données de mots de toutes les classes grammaticales ont donc été bâties à partir de zéro. En conséquence, les données créées pour ce projet sont loin d'être exhaustives. Elles sont en fait limitées aux mots qui ont été repérés dans les corpus de référence choisis, ainsi qu'à une liste de mots courants de la langue française, identifiés dans divers documents. Les sous-sections suivantes (Sections 3.1.3.1 à 3.1.3.3) portent sur la création de ces banques de données pour le projet actuel pour les différentes classes grammaticales.

Tableau 3.8 : Exemples de lemmes

Classe	Exemples	Forme lemmatisée
Verbes	marche, marchais, marcherai, marchai, etc.	marcher (infinitif)
Adjectifs	bleu, bleus, bleue, bleues	bleu (masculin singulier)
	beau, beaux, belle, belles	beau (masculin singulier)
Noms	chaise, chaises	chaise (singulier)
	instructeur, instructeurs, instructrice, instructrices	instructeur (masculin singulier)
Adverbes	lentement	lentement (pas de forme fléchie)
Déterminants	le, la, les	le (masculin singulier)
	ce, ces, cette	ce (masculin singulier)
Pronoms	il, elle	conservés tels quels, par choix
Prépositions, conjonctions, interjections	pas de formes fléchies	conservées telles quelles

3.1.3.1. Les verbes

La langue française comporte un très grand nombre de verbes. Heureusement, une liste quasi exhaustive de ceux-ci est fournie dans le Bescherelle (Bescherelle, 2012). On retrouve en premier lieu dans ce livre de référence plusieurs milliers de verbes à l'infinitif en ordre alphabétique, accompagnés de leur modèle de conjugaison. La Figure 3.1 fournit un exemple de verbes à l'infinitif apparaissant dans le Bescherelle. Pour chaque verbe infinitif listé, apparaît à sa droite un numéro de modèle. C'est à ce numéro de modèle qu'il faut ensuite se référer pour générer les formes fléchies ou conjuguées. En plus du modèle, des codes additionnels sont fournis, par exemple « I », « T » ou « P ». Ces codes, qui se réfèrent à l'emploi du verbe dans les phrases, ne sont pas importants pour la lemmatisation des textes. Ils ne sont donc pas discutés davantage ici. Cependant, ces codes jouent un rôle critique pour l'Étape 2 du présent projet consistant à générer des phrases. Ils sont donc discutés plus en détail à la Section 3.5. Finalement, certains verbes sont aussi associés à certaines prépositions. Ces prépositions ne sont pas utiles pour l'étape de la lemmatisation, mais joueront là aussi, un rôle critique pour la génération des phrases aléatoires à l'Étape 2. On s'y attardera donc aussi à la Section 3.5. La première banque de données de verbes pour ce projet contient donc cette liste de verbes à l'infinitif, accompagnés des codes, prépositions et modèles de verbes. Les formats exacts de ces tables et les algorithmes mis en place sont discutés au Chapitre 4.

On retrouve aussi dans le Bescherelle les tableaux de conjugaison pour tous les temps de verbes, correspondant aux modèles décrits plus haut, et apparaissant à la Figure 3.1 pour chaque verbe. La Figure 3.2 fournit en exemple le contenu du tableau de conjugaison du verbe « aimer », dans un format très semblable à ce qu'on retrouve dans le Bescherelle. En plus du verbe « aimer » lui-même, ce tableau indique comment tous les verbes associés à ce modèle (numéro 6 dans ce cas précis) se conjuguent. Chaque forme fléchie est alors formée par le radical du verbe d'intérêt, suivi de la terminaison fournie dans le tableau de conjugaison.

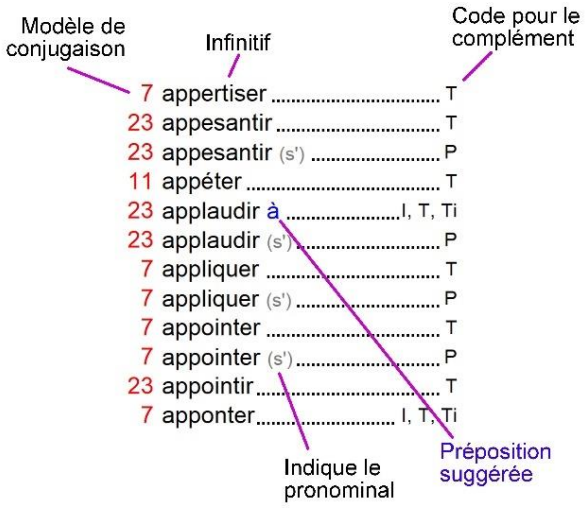


Figure 3.1 : Exemples de verbes à l'infinitif dans le Bescherelle

7 — aimer

INDICATIF

Présent		Passé composé		
j'	aime	j'	ai	aimé
tu	aimes	tu	as	aimé
elle	aime	elle	a	aimé
nous	aimons	nous	avons	aimé
vous	aimez	vous	avez	aimé
ils	aiment	ils	ont	aimé

Imparfait		Plus-que-parfait		
j'	aimais	j'	avais	aimé
tu	aimais	tu	avais	aimé
elle	aimait	elle	avait	aimé
nous	aimions	nous	avions	aimé
vous	aimiez	vous	aviez	aimé
ils	aimaient	ils	avaient	aimé

Passé simple		Passé antérieur		
j'	aimai	j'	eus	aimé
tu	aimas	tu	eus	aimé
elle	aima	elle	eut	aimé
nous	aimâmes	nous	eûmes	aimé
vous	aimâtes	vous	eûtes	aimé
ils	aimèrent	ils	eurent	aimé

Futur simple		Futur antérieur		
j'	aimerai	j'	aurai	aimé
tu	aimeras	tu	auras	aimé
elle	aimera	elle	aura	aimé
nous	aimerons	nous	aurons	aimé
vous	aimerez	vous	aurez	aimé
ils	aimeront	ils	auront	aimé

Conditionnel présent		Conditionnel passé		
j'	aimerais	j'	aurais	aimé
tu	aimerais	tu	aurais	aimé
elle	aimerait	elle	aurait	aimé
nous	aimerions	nous	aurions	aimé
vous	aimeriez	vous	auriez	aimé
ils	aimeraient	ils	auraient	aimé

SUBJONCTIF

Présent		Passé		
que j'	aime	que j'	aie	aimé
que tu	aimes	que tu	aies	aimé
qu'elle	aime	qu'elle	ait	aimé
que nous	aimions	que nous	ayons	aimé
que vous	aimiez	que vous	ayez	aimé
qu'ils	aiment	qu'ils	aient	aimé

Imparfait		Plus-que-parfait		
que j'	aimasse	que j'	eusse	aimé
que tu	aimasses	que tu	eusses	aimé
qu'elle	aimât	qu'elle	eût	aimé
que nous	aimassions	que nous	eussions	aimé
que vous	aimassiez	que vous	eussiez	aimé
qu'ils	aimassent	qu'ils	eussent	aimé

IMPERATIF

Présent	Passé	
aime	aie	aimé
aimons	ayons	aimé
aimez	ayez	aimé

INFINITIF

Présent	Passé
aimer	avoir aimé

PARTICIPE

Présent	Passé (composé)
aimant	ayant aimé
	Passé
	aimé

Figure 3.2 : Tableau de conjugaison du verbe « aimer ». Format semblable à celui du Bescherelle

Pour les besoins de la lemmatisation, les temps de verbes composés, soit le passé composé, le plus-que-parfait, le passé antérieur, le futur antérieur, le subjonctif passé, le subjonctif plus-que-parfait, l'impératif passé et les deux formes du conditionnel passé, ne sont pas considérés. En effet, ces formes composées sont bâties à partir soit de l'auxiliaire « avoir » soit de l'auxiliaire « être », suivi du participe passé. Comme les auxiliaires « avoir » et « être » sont reconnus et classifiés en tant que tel par l'outil de lemmatisation, il n'est pas nécessaire de les inclure par le biais des formes composées de tous les autres verbes. Le format exact de ces tables de conjugaisons et les algorithmes mis en place pour générer toutes les formes conjuguées des verbes sont aussi discutés au Chapitre 4.

3.1.3.2. Les noms, adjectifs et adverbes

Les noms, adjectifs et adverbes, même dans leurs formes de base (lemmes) sont extrêmement nombreux, et en établir une liste exhaustive aurait été bien trop ambitieux pour ce projet. À la place, des listes pour ces trois classes grammaticales ont été bâties à partir des mots retrouvés dans des corpus de référence comportant plus d'une centaine de milliers de mots. Les corpus de référence choisis sont présentés à la Section 3.4.

Des tables distinctes ont été bâties pour deux catégories parmi lesquelles les noms communs ont été classés, soit les noms « animés », et les noms « non-animés ». Les noms animés sont ceux que l'on peut retrouver aux deux genres et aux deux nombres, pour un total de quatre formes possibles (masculin singulier, masculin pluriel, féminin singulier et féminin pluriel). On les dit « animés » car ils font souvent, mais pas toujours, référence à des personnes, donc des êtres « animés ». Par exemple, on retrouve « infirmier », « infirmiers », « infirmière » et « infirmières ». Les noms dits « non-animés » sont ceux qu'on ne retrouve qu'à un seul genre. Par exemple, on dit « une chaise » et non « un chaise ». Ces noms peuvent aussi généralement se retrouver autant au singulier qu'au pluriel. Il existe certains mots cependant, qu'on ne retrouve qu'au singulier, ou qu'au pluriel. Par exemple, le Petit Robert dico en ligne (2024) mentionne que certains noms de matière (« farine », « café », « soie ») ou abstraits (« peur », « gentillesse ») s'emploient presque exclusivement au singulier. Mais on peut tout de même les retrouver au pluriel dans certains contextes. On retrouve plus souvent des noms dont on ne se sert qu'au pluriel. Le Petit Robert dico en ligne fournit là encore quelques exemples : « décombres », « fiançailles », « funérailles », « mœurs ». Dans les banques de données pour ce projet, ces mots employés uniquement au pluriel sont proprement identifiés. Bien que non critique pour la lemmatisation, cette identification s'avère utile pour l'Étape 2 du projet, pour s'assurer de ne pas utiliser au singulier un mot ne s'employant qu'au pluriel, ou vice-versa, lors de la génération de phrases aléatoires.

Les adverbes quant à eux, sont invariables et n'ont donc pas de formes fléchies. Tout comme pour les noms et adjectifs, une liste a été repérée au sein des corpus de référence.

3.1.3.3. Les déterminants, pronoms, prépositions, conjonctions et interjections

Les mots appartenant à ces classes grammaticales apparaissent très fréquemment dans les textes, mais leurs formes distinctes ne sont pas nombreuses. Il a en effet été possible d'extraire une liste quasi exhaustive de tous les déterminants, pronoms, prépositions et conjonctions. Pour ce qui est des interjections, celles-ci peuvent être très nombreuses, car souvent à la merci de la créativité des auteurs. Certaines d'entre elles, celles retrouvées dans les corpus de référence, ont été incorporées aux banques de données.

3.1.4. Préparation du corpus de référence

La lemmatisation d'un corpus de référence requiert qu'on analyse, un à un, chaque mot le composant. La méthode détaillée pour y arriver est discutée à la Section 3.1.5. Mais en préparation pour cette analyse, il faut d'abord simplifier le corpus, pour en retirer les informations non nécessaires à l'étape présente. Par exemple, bien que la ponctuation nous soit utile ultérieurement pour la désambiguïsation des homographes (Section 3.2), elle n'est pas nécessaire pour la lemmatisation en tant que tel. De la même façon, les majuscules en début de phrase ne sont pas pertinentes non plus, car qu'un mot apparaisse au début de la phrase ou ailleurs dans celle-ci, il s'agit toujours du même mot. Les majuscules servent aussi à identifier les noms propres, mais la méthode utilisée pour distinguer ceux-ci est abordée uniquement au Chapitre 4 (Section 4.2.4), portant sur les algorithmes informatiques développés pour le projet. De la même façon, les algorithmes servant à « nettoyer » le texte en préparation pour la lemmatisation sont aussi détaillés au Chapitre 4. On ne se soucie pour l'instant que de mentionner le besoin de préparer le corpus. Cette étape de préparation, ou nettoyage, est illustrée à la Figure 3.3. On y observe un exemple de version originale d'un texte à gauche, puis sa version nettoyée à droite, qui sert par la suite à la lemmatisation.

3.1.5. Méthodologie pour la lemmatisation

Une fois les banques de données formées pour toutes les classes grammaticales (Section 3.1.3) et le corpus de référence « nettoyé » (Section 3.1.4), les ingrédients sont là pour débuter le processus de lemmatisation du corpus de référence. Il faut se rappeler, tel que mentionné au début de ce chapitre (Section 3.1), que l'objectif de l'Étape 1 est de générer une liste de verbes, noms, adjectifs et adverbes représentatifs du corpus de référence avec leurs fréquences d'apparition dans le texte. On extrait aussi certaines autres statistiques de base en lien avec le corpus, telles que la fréquence des temps de verbes et la fréquence des personnes utilisées pour les verbes. Ces informations sont par la suite utilisées pour la sélection des paramètres pour la génération de textes aléatoires automatiquement lemmatisés à l'Étape 2. L'approche choisie pour extraire cette information est de concevoir un lemmatiseur, c'est-à-dire un outil qui associe chaque mot du texte de référence avec son lemme de base. Une partie des statistiques requises est obtenue via l'analyse de ces lemmes. La méthode employée pour créer ce lemmatiseur est l'objet de la présente section.

L'algorithme de lemmatisation développé pour ce projet s'attarde à chaque mot du corpus, un à la suite de l'autre. Prenant l'exemple du texte nettoyé de la Figure 3.3, le premier mot à analyser est « les ». L'algorithme compare ce mot à tous les mots des banques de données discutées à la Section 3.1.3, pour tenter de l'y repérer.

Texte original	Texte nettoyé
Les gros camions qui roulent sur l'autoroute doivent s'immobiliser, le temps d'être pesés et inspectés. Ils pourront ensuite poursuivre leur route, sans aucun autre souci.	les gros camions qui roulent sur l autoroute doivent s immobiliser le temps d être pesés et inspectés ils pourront ensuite poursuivre leur route sans aucun autre souci

Figure 3.3 : Étape de préparation (nettoyage) du texte pour la lemmatisation

Pour ce mot en particulier, on retrouve deux occurrences dans les banques de données, soit le déterminant « les » et le pronom « les »[1]. Bien que dans ce contexte précis de la phrase analysée il s'agisse du déterminant et non du pronom, l'algorithme de lemmatisation ne se préoccupe pas pour l'instant de cette distinction. On y reviendra à la Section 3.2, portant sur la désambiguïsation des homographes. En effet, l'algorithme conserve pour l'instant tous les lemmes et classes grammaticales associées à chaque mot rencontré. Dans un cas comme dans l'autre (déterminant ou pronom), le lemme associé au mot « les » est « le ». On passe ensuite au mot suivant, « gros ». Une fois de plus, on compare ce mot à tous ceux présents dans les banques de données, et on y repère trois lemmes, soit l'adjectif « gros », le nom « gros » ou l'adverbe « gros ». Bien que dans chacun de ces trois cas le lemme partage la même orthographe (« gros »), trois classes grammaticales entrent en jeu. Le lemmatiseur conserve donc en mémoire ces trois possibilités, ne cherchant pas pour l'instant à déterminer laquelle de ces trois options est ici la seule pertinente (l'adjectif dans la phrase présente). On passe ensuite au mot suivant, « camions ». La recherche parmi les banques de données nous mène au lemme « camion », un nom commun. Dans ce cas précis, on ne repère que ce seul lemme, qui sera donc le seul enregistré pour ce mot. L'algorithme poursuit ainsi l'analyse avec tous les autres mots du corpus. Les résultats obtenus à cette étape sont illustrés au Tableau 3.9, pour la première phrase du texte de la Figure 3.3.

Tableau 3.9 : Résultat de la lemmatisation d'une phrase

Mot	Lemmes et classes grammaticales
les	les (déterminant), les (pronom)
gros	gros (adjectif), gros (nom commun), gros (adverbe)
camions	camion (nom commun)
qui	qui (pronom relatif), qui (pronom interrogatif)
roulent	rouler (verbe)
sur	sur (préposition), sur (adjectif)
l	le (déterminant), la (déterminant), les (déterminant), le (pronom), la (pronom), les (pronom)
autoroute	autoroute (nom commun)
doivent	devoir (verbe)
s	se (pronom personnel)
immobiliser	immobiliser (verbe)
le	le (déterminant), le (pronom)
temps	temps (nom commun)
d	de (déterminant), de (pronom)
être	être (nom), être (verbe)
pesés	peser (verbe)
et	et (conjonction)
inspectés	inspecter (verbe)

[1] Le mot « les » peut aussi servir de préposition signifiant « près de », utilisée dans certains noms de lieu, comme « Tremblant-les-Bains ». Son orthographe est parfois « lez » ou « lès ». Cette utilisation n'est pas considérée ici.

On voit donc que l'algorithme, pour chaque mot, en identifie toutes les classes grammaticales possibles ainsi que les lemmes correspondants. On constate au Tableau 3.9 que certains mots n'ont qu'un seul lemme qui leur est associé. Ces mots sont donc considérés comme étant « non ambigus ». Aucune analyse subséquente n'est ainsi nécessaire pour ces mots non ambigus pour en déterminer le lemme et la fonction précise dans la phrase. En revanche, tous les mots pour lesquels plus d'un lemme ou d'une classe grammaticale ont été repérés peuvent être considérés comme étant des *homographes*, c'est-à-dire des mots dont la graphie est la même, mais dont le lemme et/ou la classe grammaticale diffèrent. La définition précise du terme « homographe » utilisée dans ce projet est discutée plus en détail à la Section 3.2.2. Les algorithmes informatiques utilisés pour emmagasiner l'information sont quant à eux décrits au Chapitre 4. L'algorithme ne se préoccupe pas pour l'instant d'un étiquetage plus précis de chaque mot rencontré pour en déterminer par exemple le genre, le nombre, la personne ou le temps, selon la classe grammaticale en jeu. Cette information est considérée plus tard.

D'autres informations concernant le corpus de référence sont aussi extraites en parallèle à l'étape présente. En effet, bien que la ponctuation ait été retirée lors du nettoyage du texte (Section 3.1.4), une version non nettoyée du texte sert aussi à déterminer le début et la fin de chaque phrase du corpus. Cette information sert à l'étape de la désambiguïsation des homographes (Section 3.2), puisque celle-ci s'effectue au niveau de la phrase. Il est donc essentiel de bien délimiter chacune des phrases pour l'étape subséquente du traitement des homographes. Le lemmatiseur se sert aussi de la position du début de chaque phrase pour extraire les statistiques sur les longueurs des phrases, pour des fins de comparaison avec les textes automatiquement lemmatisés qui seront générés à l'Étape 2 de ce travail.

3.2. Désambiguïsation des homographes

La désambiguïsation des homographes représente un aspect critique de tout outil de lemmatisation. En effet, le but ultime de tels outils, généralement, est d'associer chaque mot du texte à un et un seul lemme, celui qui est pertinent dans le texte. Tel que le mentionne Hug (2002), un programme informatique ne peut procéder comme un humain « par intuition, en utilisant sa compétence de locuteur ». Au contraire, un outil informatique doit se baser sur des règles précises, qui néanmoins introduisent « une proportion plus ou moins importante d'analyses erronées ». La présente section a pour but d'illustrer la méthode de désambiguïsation développée ici pour optimiser le processus, et donc minimiser ces analyses erronées. Mais en premier lieu, il est bon de discuter de sa pertinence dans le projet actuel (Section 3.2.1) et par la suite de définir ce qu'on entend exactement par le terme « homographe » (Section 3.2.2), car les définitions et surtout l'interprétation de ce terme peuvent varier et influer sur les algorithmes à développer.

3.2.1. Pertinence de la désambiguïsation dans le projet actuel

Tel que le mentionne Hug (2002), l'intérêt de la désambiguïsation varie selon le but qu'on poursuit. Si l'objectif visé est sémantique, l'intérêt de la désambiguïsation est variable. Hug cite d'abord en exemple les homographes « cause », « analyse », « bagarre », « contraste » et « couronne ». Pour ceux-ci, les contenus sémantiques du nom et du verbe étant très proches, l'intérêt de la désambiguïsation est faible. Au contraire, pour les formes « laisse », « ferme » ou « bouche » que cite ensuite Hug, la forme verbale et le nom n'évoquant pas du tout le même sens, la désambiguïsation demeure pertinente quand l'objectif visé est d'ordre sémantique.

Mais pour juger de la pertinence de la désambiguïsation dans le cas du projet actuel, il faut une fois de plus revenir à l'objectif visé. L'Étape 1 de ce projet, tel que mentionné plus tôt, consiste à générer des banques de données de mots *représentatifs d'un corpus de référence*. On cherche donc à quantifier les fréquences d'apparition de ces mots, de même que certaines statistiques en lien avec le corpus. De plus, ce n'est que pour les verbes, les noms, les adjectifs et les adverbes

qu'on s'intéresse ici aux fréquences d'occurrences. On ne poursuit donc pas pour le projet actuel, comme but premier, une lemmatisation parfaite du corpus.

De ce point de vue, l'étape de désambiguïsation peut même sembler futile. On pourrait en effet choisir, à chaque fois qu'on croise un homographe, de compiler dans nos statistiques de fréquences toutes les formes de l'homographe, sans se soucier de laquelle est la bonne dans le contexte utilisé. Par exemple, dans la portion de phrase « la *dame* en bleu », bien que le mot « dame » soit ici un nom commun, l'algorithme pourrait considérer de façon égale la possibilité que ce mot soit plutôt une forme conjuguée du verbe « damer ». Dans le contexte de ce projet, la conséquence d'erronément considérer le verbe « damer » lorsque ce n'est pas le cas, est que nous allons générer des phrases aléatoires à l'Étape 2 dans lesquelles on retrouvera le verbe « damer » plus fréquemment que ne nous le suggérerait sa fréquence réelle d'apparition dans le corpus. De plus, le verbe « damer » pourrait se retrouver à être conjugué à d'autres formes que celles correspondant aux graphies « dame » ou « dames ». Cette conséquence n'est pas critique, considérant que le but ultime poursuivi est de générer des phrases aléatoires automatiquement lemmatisées. Les phrases bâties avec le verbe « damer » seraient en effet tout de même parfaitement lemmatisées et construites en parfait accord avec les règles du français. La seule conséquence fâcheuse, est qu'on s'éloignerait ainsi du lexique du corpus de référence, en choisissant des termes qui n'auraient pas été rendus disponibles avec une désambiguïsation bien effectuée.

C'est donc initialement dans le contexte d'un objectif secondaire du projet, celui de générer des phrases *représentatives du corpus de référence*, que la désambiguïsation prend ici tout son sens. Plus la désambiguïsation est efficace, plus les phrases générées aléatoirement sont représentatives. La perfection n'est donc pas critique, seulement souhaitable.

Mais le développement ici d'un outil de désambiguïsation des homographes contribue aussi directement à l'objectif principal de ce travail qui concerne l'évaluation d'outils de lemmatisation existants. En effet, un effort soutenu en vue de désambiguïser les homographes met en relief les défis et difficultés reliés à ce processus. On peut alors confronter les outils de lemmatisation existants à des textes comportant des mots et expressions identifiés comme difficiles à désambiguïser ou à ce qu'on pourrait considérer comme étant des « pièges ».

C'est donc en gardant en tête ces deux objectifs, que nous discutons maintenant des approches choisies pour la désambiguïsation des homographes.

3.2.2. Définition choisie pour le concept d'homographe

Le dictionnaire Usito (2024) fournit la définition suivante pour le terme « homographe » : « mot qui a la même orthographe, et parfois la même prononciation qu'un autre mais un sens différent. ». Mais dans le contexte précis de ce projet de recherche, on propose plutôt la définition suivante :

> Homographe : Mot qui a la même orthographe qu'un autre, mais dont le lemme et/ou la classe grammaticale diffèrent.

On constate qu'on a ici retiré la référence à la prononciation, qui n'est pas pertinente dans le projet actuel, qui ne porte que sur l'écrit. On constate aussi qu'on ne fait pas référence au sens, puisque les algorithmes développés au cours de ce projet n'ont pas été conçus pour analyser la sémantique et que les définitions des mots n'ont pas été incluses dans les banques de données. La définition du terme « homographe » proposée ici se concentre donc sur deux autres aspects, soit le lemme et la classe grammaticale. Des exemples sont fournis au Tableau 3.10 pour bien mettre la définition proposée en contexte.

Selon la définition proposée plus haut pour ce mémoire et l'analyse du Tableau 3.10, on en déduit que les mots « gros », « voie » et « fils » sont bel et bien des homographes. Et le corolaire de la

définition proposée plus haut est que si deux mots ont le même lemme et la même classe grammaticale, ils ne sont *pas* des homographes. Il vaut toutefois la peine de s'attarder à certains de ces cas pour bien clarifier la définition adoptée ici, ce qui est fait au Tableau 3.11.

Dans le cas du nom commun « verre », on constate au Tableau 3.11 que même si on a affaire à un seul lemme et une seule classe grammaticale, il y a plusieurs sens possibles pour ce nom. En se fiant à la définition fournie par le dictionnaire Usito, on en déduirait donc que « verre » *est* un homographe. Mais selon la définition adoptée pour ce travail, le mot « verre » n'est pas considéré comme un homographe, car le lemme et la classe grammaticale demeurent les mêmes, peu importe le sens du mot dans le contexte où il est utilisé. En effet, l'algorithme de désambiguïsation développé ici ne cherchera pas à distinguer les mots dont uniquement le sens diffère. Autrement dit, bien que le dictionnaire Usito considère le mot « verre » comme étant un homographe, ce n'est pas le cas en utilisant la définition adoptée pour ce projet de recherche.

Pour le deuxième exemple du Tableau 3.11 (« aime »), il n'est pas clair quel serait le constat selon la définition du mot « homographe » fournie par le dictionnaire Usito. Tout d'abord, toutes les conjugaisons possibles du verbe « aimer » menant à la forme « aime » s'écrivent de la même façon et se prononcent de la même façon, ce qui rencontre les deux premières conditions fournies par Usito.

Tableau 3.10 : Homographes selon le lemme et la classe grammaticale

Lemme	Classe grammaticale	Exemples
Même	Différente	« gros » - peut être un nom, un adjectif ou un adverbe
Différent	Différente	« voie » - peut être un nom (lemme = « voie »), ou un verbe (lemme= « voir »)
Différent	Même	« fils » - nom commun, pouvant être associé au lemme « fil » (brin de matière), ou au lemme « fils » (personne de sexe masculin) « suis » - verbe, pouvant être associé au lemme « être », ou au lemme « suivre »

Tableau 3.11 : Mots non considérés comme homographes pour ce projet

Lemme	Classe grammaticale	Exemples
Même	Même	« verre » - nom commun. Peut signifier un récipient, une lentille ou une matière
Même	Même	« aime » - verbe « aimer ». On peut retrouver ce verbe conjugué à différents temps et différentes personnes (j'aime, il aime, que j'aime, qu'il aime, aime!)

Mais il est plus difficile de débattre du *sens* du mot « aime ». On pourrait débattre du fait que la forme conjuguée « aime », qu'elle soit au présent ou au subjonctif, à la première ou à la troisième personne, se réfère au même sens, celui d'éprouver de l'affection. Au contraire, on pourrait statuer que le sens diffère, car le mode peut être différent (indicatif ou subjonctif) ou que la personne diffère. Mais ces questions ne se posent pas si on utilise la définition d'homographe choisie pour ce travail. En effet, selon la définition adoptée ici, il est clair que ces cinq formes conjuguées (« aime ») du verbe aimer ne sont *pas* des homographes, puisqu'on fait référence au même lemme (« aimer ») et à la même classe grammaticale (verbe). Il est toutefois à noter que les formes conjuguées identiques d'un même verbe seront traitées de façon particulière dans ce projet de recherche. Mais une discussion plus approfondie à ce sujet n'a lieu qu'au Chapitre 4.

Si l'on revient au Tableau 3.9 portant sur la lemmatisation d'une phrase en particulier, on constate que plusieurs mots ont été associés à plus d'un lemme et/ou plus d'une classe grammaticale possibles. Considérant la définition du concept d'homographe adoptée pour ce projet, on en déduit donc que ces mots sont effectivement considérés comme des homographes. En effet, leur graphie seule ne peut servir à déterminer sans ambiguïté leur lemme et/ou leur classe grammaticale. Il est à noter que le mot « doivent » correspond à deux formes conjuguées du verbe « devoir », soit à la troisième personne du pluriel du présent de l'indicatif, et du subjonctif présent. Mais tel que discuté plus haut, on ne parle pas ici d'homographe, car dans ces deux cas, le lemme et la classe grammaticale demeurent les mêmes.

La présence d'homographes de classes grammaticales différentes n'est pas unique au français. Par exemple, Hein (1990) mentionne qu'environ 85% des homographes suédois proviennent de classes de mots différents, par exemple un nom et un verbe. En français, comme on le confirmera au Chapitre 5 en fonction des corpus de référence, la grande majorité des homographes implique aussi des classes grammaticales différentes. Les homographes appartenant à la même classe grammaticale sont plus rares. Parmi eux, on retrouve par exemple en français des formes de verbes conjugués identiques, mais se référant à un infinitif différent. Par exemple « je suis » issu du verbe être, et « je suis » issu du verbe « suivre ». Deux lemmes sont donc possibles pour le mot « suis », soit « être » ou « suivre ». Parmi les noms communs, on retrouve certains cas où la forme au pluriel d'un nom est la même que la forme au singulier d'un autre, par exemple pour le mot « fils ». En effet, ce mot peut être le pluriel de « fil » (brin de tissu) ou plutôt la forme au singulier « fils », synonyme d'enfant de sexe masculin.

Le Tableau 3.12 fournit plusieurs exemples d'homographes en fonction de certaines des combinaisons possibles de classes grammaticales définies au Tableau 3.1. Les cases en rose représentent les cas d'homographes appartenant à la même classe grammaticale, tandis que les cases en blanc concernent les homographes appartenant à deux classes différentes. On constate que certaines cases sont vides. C'est qu'il n'y a pas d'exemples d'homographes, selon la définition adoptée pour ce projet, dans ces catégories, du moins au sein des corpus de référence choisis. Par exemple, il n'y a aucun déterminant qui puisse aussi être une forme verbale. Et comme la liste de déterminants est somme toute assez réduite, c'est là une vérification facile à effectuer. En ce qui a trait aux homographes appartenant à la même classe grammaticale, on n'a pu en trouver ni pour les adverbes, ni pour les déterminants et ni pour les adjectifs.

Les deux sections suivantes (Sections 3.2.3 et 3.2.4) discutent de stratégies pour la désambiguïsation d'homographes selon justement qu'ils appartiennent à des classes grammaticales différentes, ou à la même classe.

Tableau 3.12 : Exemples d'homographes selon leurs classes grammaticales

verbe	adjectif	nom	adverbe	déterminant	pronom	
« suis » *(être ou suivre)*	« évident » *(verbe « évider »)*	« laisse »	« maintenant » *(verbe « maintenir »)*		« tienne »	verbe
		« rose »	« gros »	« certain »	« certain »	adjectif
		« fils »	« pas »	« ton »	« rien »	nom
				« tout »	« tout »[2]	adverbe
					« le »	déterminant
					« quoi » *(interrogatif ou relatif)*	pronom

3.2.3. Désambiguïsation d'homographes ayant une classe grammaticale différente

Tel que mentionné plus haut, le but ultime d'un outil de lemmatisation est d'associer chaque mot d'un texte à un et un seul lemme, celui qui est pertinent dans le texte. En présence d'homographes tels que ceux listés à la Section 3.2.2, un algorithme de lemmatisation doit donc déterminer laquelle des formes d'un homographe est celle à conserver. Pour y arriver, il existe différentes stratégies. Celles-ci diffèrent selon le type d'homographes auquel on fait affaire. La section présente se concentre sur les stratégies de désambiguïsation à adopter dans le cas où les homographes en jeu appartiennent à des classes grammaticales *différentes*.

3.2.3.1. Séquence des mots dans la phrase

A chaque classe grammaticale, on associe une fonction différente dans la phrase. Par exemple, un déterminant précède le nom car il l'introduit. Un adjectif quant à lui, précise le sens du nom auquel il se rapporte. Il peut soit précéder le nom (adjectif antéposé – « le gros camion ») ou le suivre (postposé – « le camion rouge »). Le pronom, qui remplace le nom dans un groupe nominal, ne peut être situé n'importe où dans la phrase. De ces observations, on constate que l'ordre des mots dans une phrase n'est jamais complètement aléatoire. La position d'un homographe dans une phrase, en comparaison avec les autres mots qui la composent, offre donc des indices quant à la classe à laquelle il appartient. Une première stratégie de désambiguïsation des homographes de classes grammaticales différentes consiste donc à étudier la séquence des mots dans la phrase.

Comme premiers indices pour faciliter la désambiguïsation d'homographes, la classe du mot précédant l'homographe, et la classe du mot suivant l'homographe seront considérées. Il n'existe pas de séquence fixe de classes grammaticales dans une phrase bien bâtie. Par exemple, un déterminant ne précède pas toujours un nom. En effet, un adjectif antéposé peut se glisser entre le déterminant et le nom commun. De la même façon, l'adjectif peut se retrouver soit avant ou après le nom qu'il modifie. Bien qu'un adverbe se retrouve souvent tout juste après un verbe, ce n'est pas non plus toujours le cas. De plus, les mots appartenant à certaines classes grammaticales ont plus de chances de se retrouver en tout début de phrase ou en toute fin, ou de

[2] Le mot « tout », très polyvalent, peut être un déterminant (« tout le monde »), un adjectif (« de tout cœur »), un adverbe (« tout nu »), un pronom indéfini (« tout est fini ») ou un nom commun (« former un tout »)

part et d'autre d'une virgule ou d'un trait d'union. Il n'y a donc pas de règle fixe quant à l'ordre des mots, sans pour autant que l'ordre soit complètement aléatoire.

Aux indices reliés à la classe grammaticale, peuvent s'ajouter des indices un peu plus précis concernant la « sous-classe » grammaticale de certains mots. Par exemple, on peut vérifier si l'homographe qu'on tente de désambiguïser est précédé ou suivi d'un pronom démonstratif (cela, celle, celui), d'un adverbe spatio-temporel (autour, dehors, auparavant, autrefois), d'un participe passé, ou d'un nom propre.

Une analyse encore plus précise des mots environnant l'homographe sous étude peut parfois apporter des indices additionnels. Il est fort probable par exemple que suivant les mots « du », « de la » ou « des », on retrouve un nom commun. Certains autres mots comme « en », « ne », « pas », « et », « plus », « moins » ou « si », peuvent aussi servir d'indices judicieux pour désambiguïser les homographes dans leur environnement immédiat.

Ainsi, comme la séquence des mots dans une phrase n'est pas complètement aléatoire, une approche statistique sera utilisée pour évaluer la *probabilité* qu'un homographe appartienne à une classe en particulier, en fonction des mots qui le précèdent et qui le suivent, sur la base des indices listés plus haut. Cette approche, basée sur l'apprentissage machine, sera détaillée au Chapitre 4.

3.2.3.2. Application des règles d'accord grammatical de base

Un autre aspect important de certaines classes de mots, est le fait que les mots qui les composent ont des formes variables dues à certaines de leurs caractéristiques. Par exemple, à un nom ou un adjectif, on associe un genre et un nom. À un déterminant, on associe la plupart du temps aussi un genre et un nombre, et parfois aussi une personne, comme dans le cas de déterminants possessifs. Même chose pour les pronoms, mais en plus, ceux-ci sont associés à un « cas » : nominal, accusatif ou datif, tel que décrit à la Section 3.1.1. Les verbes quant à eux, sont associés à un temps et une personne.

À ces caractéristiques, on associe ensuite certaines règles de grammaire. Par exemple, les déterminants doivent toujours s'accorder en genre et en nombre avec le nom ou les noms qu'ils déterminent. Les adjectifs doivent aussi s'accorder en genre et en nombre avec le nom qu'ils modifient. Et les verbes doivent se conjuguer en fonction de leur sujet, en adoptant la même personne.

Ces règles de grammaire peuvent apporter des indices pertinents pour la désambiguïsation des homographes. En effet, pour considérer la possibilité qu'un mot soit un déterminant, on doit localiser à sa suite un nom auquel il se rapporte et ensuite s'assurer que ce nom est du même genre et nombre que le déterminant en question. Si c'est le cas, la possibilité que l'homographe en question soit un déterminant est forte. Prenons en exemple la courte phrase suivante :

> « Je te le montre. »

Concentrons-nous sur le mot « le ». De façon générale, ce mot peut-être soit un déterminant, soit un pronom, et nous cherchons à déterminer laquelle de ces deux possibilités est pertinente ici. Si on émet l'hypothèse que « le » est un déterminant, on cherche la présence d'un nom à sa suite. On trouve le mot « montre », qui peut être soit un nom ou un verbe. Il semble donc possible que le mot « le » soit un déterminant. Cependant, il faut ensuite vérifier que le déterminant s'accorde avec le nom. Et ici, comme « le » est masculin et que le nom « montre » est féminin, on voit que la règle des accords ne serait pas respectée. On peut en déduire qu'ici, le mot « le » ne peut pas être un déterminant. Par élimination, il est donc un pronom. La même conclusion aurait été atteinte avec la phrase « Je te les montre ».

Toutefois, une telle analyse n'est pas toujours aussi facile. Si la phrase avait été « Je te la montre », on n'aurait pas pu utiliser cet unique argument, puisque dans un tel cas, le déterminant « la » et le nom « montre » sont tous les deux féminins et singuliers, malgré le fait que le mot « la » ici est un pronom. Davantage d'indices sont alors nécessaires.

Pour permettre la désambiguïsation d'homographes de classes grammaticales différentes, il est donc utile de vérifier si les règles grammaticales (accords) sont respectées en fonction des homographes possibles. Ainsi, certaines règles grammaticales seront considérées pour la désambiguïsation d'homographes. L'utilisation de ces règles de grammaire est double. Comme le mentionnent Vergne et Giguet (1998), on peut faire appel à des « déductions négatives » lorsqu'une règle de grammaire fait en sorte qu'on arrive à supprimer la possibilité d'un homographe. Au contraire, on appelle « déduction affirmative » quand une règle confirme qu'un homographe en particulier puisse être utilisé dans le contexte de la phrase sous étude. Des règles d'accord en genre et en nombre utilisées pour ce projet nous permettant de faire de telles déductions, sont listées au Tableau 3.13.

Comme plusieurs homographes impliquent des formes verbales, il est aussi essentiel de rechercher des indices aidant à déterminer si un homographe peut en effet être un verbe dans le contexte de la phrase étudiée. Par exemple, on peut profiter du fait qu'on retrouve généralement un auxiliaire devant un participe passé (« j'*ai* mangé »). On retrouve plutôt parfois un verbe attributif devant un participe passé (« il *demeure* aimé »). Aussi, on sait que quand deux verbes se suivent, le deuxième est à l'infinitif (sauf pour un auxiliaire suivi d'un participe passé). Ainsi, si deux mots qui sont potentiellement des verbes conjugués se suivent, il est fort à parier qu'au moins l'un des deux n'est pas un verbe. Par exemple, dans la phrase :

> « La montre indique l'heure. »

Le mot « montre » pourrait être soit un nom commun ou soit un verbe conjugué. Mais comme ce mot est suivi d'un verbe conjugué (« indique »), le mot « montre » ne peut pas lui aussi être un verbe conjugué. Par élimination, il doit donc être un nom commun.

De plus, certaines prépositions exigent que le verbe qui les suit soit à l'infinitif. Ainsi, à la suite des prépositions « à », « de », « pour » et « sans », un verbe doit être à la forme infinitive (« J'ai beaucoup *à* faire »). Donc, si un homographe pouvant de façon générale être un verbe conjugué suit une telle préposition, il ne peut être un verbe. Par exemple :

> « Il n'a pas de montre. »

Dans un tel cas, on doit éliminer la possibilité que l'homographe « montre » soit un verbe, car il aurait plutôt fallu qu'on retrouve la forme infinitive (« montrer »). On en déduit qu'il s'agit ici plutôt d'un nom commun.

Tableau 3.13 : Exemples de règles d'accord en genre et nombre utilisées pour la désambiguïsation

Classe ou sous-classe grammaticale concernée	Règle de grammaire	Exemples
nom commun ou adjectif antéposé	Si précédé d'un déterminant, le mot doit s'accorder en genre et en nombre avec le déterminant	la montre le montre (erroné) la belle montre
nom commun	Si suivi d'un adjectif postposé, le nom doit s'accorder en genre et en nombre avec l'adjectif. Permettre la présence d'un adverbe entre les deux	le bijou précieux le bijou très précieux
nom commun	Si précédé d'un adjectif antéposé, le nom doit s'accorder en genre et en nombre avec l'adjectif	la belle montre
adjectif	Si précédé d'un verbe attributif, le nombre doit concorder	Ils sont beaux. Il devient sec.
adjectif antéposé	S'accorde en genre et en nombre avec le nom commun qui le suit	la belle montre
adjectif postposé	S'accorde en genre et en nombre avec le nom commun qui le précède. Permettre la présence d'un adverbe entre les deux	le bijou précieux le bijou très précieux
déterminant (non numéral)	S'accorde en genre et en nombre avec le nom commun, l'adjectif ou le déterminant numéral qui le suit	les bijoux ces beaux bijoux leurs trois bijoux

Aussi, les verbes doivent être conjugués à la personne correspondant à leur sujet. Ainsi, dans la phrase « Tu montres le chemin. », le verbe conjugué « montres » s'accorde avec le pronom personnel « tu », ce qui donne un indice comme quoi le mot « montres » est effectivement un verbe conjugué dans ce contexte. Au contraire, dans la phrase « Tu perds ta montre. », le mot « montre » ne peut pas être le verbe conjugué associé au pronom personnel « tu », car il n'est pas conjugué à la deuxième personne du singulier. Le mot « montre » a donc plus de chances d'être un nom, et c'est plutôt le mot « perds » qui est ici le verbe conjugué associé au pronom « tu ».

Finalement, dans la vaste majorité des phrases de la langue française, on retrouve au moins un verbe conjugué. Ceci peut là aussi nous donner des indices sur la classe grammaticale d'un homographe. Examinons la phrase suivante :

> « Je lui montre mon rapport ».

Dans cette phrase, le mot « montre » est un homographe, pouvant être soit un nom commun, soit un verbe conjugué. Certaines des règles précédemment mentionnées fournissent déjà des indices pour déterminer la classe grammaticale de ce mot. Mais en analysant la phrase dans son ensemble, on remarque que si le mot « montre » n'est pas un verbe, la phrase ne comporte alors aucun verbe conjugué. Sur la base de ce seul argument, il est fort à parier que le mot « montre » soit un verbe dans cette phrase. Cet indice perd cependant de son utilité dans des phrases plus

complexes comportant plus d'un verbe conjugué, et ne fonctionne pas non plus dans des courtes phrases ne comprenant pas de verbes, par exemple :

« Ma montre! Je l'ai perdue! »

Dans la première de ces deux très courtes phrases, il n'y a pas de verbe. Utiliser comme indice le fait que la phrase ne comporte aucun verbe conjugué pour déterminer que le mot « montre » doit être un verbe nous mène donc sur une fausse piste. Toujours est-il qu'ici, la présence d'un déterminant devant le mot (« ma ») fait pencher la balance vers l'assignation du mot « montre » comme un nom commun. Il faut garder en tête que nous ne parlons ici que *d'indices*, et non d'assignations absolues.

3.2.3.3. Indices additionnels favorisant la désambiguïsation des formes verbales

D'autres règles de grammaire ou de construction de phrases peuvent aussi être mises à profit pour désambiguïser certaines formes verbales. Par exemple, les formes des verbes « être » ou « avoir » sont souvent suivies de participes passés, car ces formes jouent souvent le rôle d'auxiliaires. La présence d'un participe passé, suivi directement ou non de l'homographe, fournit donc un indice pertinent.

Certaines prépositions introduisent quant à elles des formes verbales à l'infinitif, soit les prépositions « à », « de », « pour » et « sans ». Un homographe suivant une de ces prépositions a donc de fortes chances d'être un verbe à l'infinitif. Par exemple, dans le segment de phrase « il risque de devoir venir », la présence de la préposition « de » offre une forte indication que le mot « devoir » est un verbe à l'infinitif, et non le nom commun « devoir ».

Les adverbes « ne » et « pas », marquant la négation, peuvent aussi servir d'indices pour identifier certaines formes verbales. Par exemple, un homographe situé *entre* ces deux adverbes a de bonnes chances d'être une forme verbale conjuguée (« je ne *mange* pas »). Au contraire, un homographe situé après ces deux adverbes a de bonnes chances d'être une forme verbale infinitive (« il ne faut pas *manger* »).

Le Tableau 3.14 résume certaines règles grammaticales et de construction de phrase en lien avec les verbes qui sont utilisées dans ce travail, comme indices aidant à la désambiguïsation. On retrouvera au Chapitre 4 une description détaillée des algorithmes utilisés pour combiner des indices tels que ceux décrits aux Tableaux 3.13 et 3.14 pour procéder à la désambiguïsation.

Tableau 3.14 : Règles grammaticales en lien avec les verbes, utilisées pour la désambiguïsation

Sous-classe de verbe concernée	Règle de grammaire	Exemples
participe passé	Généralement précédé d'un auxiliaire être ou avoir, ou d'un verbe attributif. Les deux mots peuvent être séparés par un adverbe.	J'ai mangé Elle a trop mangé. Il demeure aimé
auxiliaire	On retrouve souvent un participe passé suite à un auxiliaire être ou avoir. Le participe passé n'est pas toujours situé directement après l'auxiliaire. Note : les verbes « être » ou « avoir » ne sont pas uniquement utilisés comme auxiliaires.	J'ai mangé. Je suis finalement arrivé.
conjugué	On ne retrouve pas deux verbes conjugués qui se suivent.	La montre indique l'heure. (« montre ») ne peut pas être un verbe conjugué ici
conjugué	Si un pronom personnel sujet (nominatif) précède un verbe, le verbe doit se conjuguer à la même personne que le pronom personnel en question. Note : La règle s'applique aussi pour les groupes nominaux employés comme sujets « les animaux mangent », mais cette règle ne sera pas appliquée pour ce projet, pour limiter la complexité des algorithmes.	Tu manges des frites. Nous regardons la télévision.
conjugué	Une phrase contient généralement au moins un verbe conjugué.	
conjugué	Dans les phrases négatives, on retrouve généralement un verbe conjugué entre les adverbes « ne » et « pas ». Il peut y avoir d'autres mots entre ces trois. D'autres adverbes jouent parfois le même rôle que « pas » : « plus », « point », « guère ».	Je ne veux pas. Elle ne croit absolument pas en lui. Tu ne veux plus venir.
infinitif	Suite aux prépositions « à », « de », « pour » et « sans », un verbe doit se retrouver à la forme infinitive.	Je dessine sans dépasser. J'ai beaucoup à faire.
infinitif	Dans une phrase négative, c'est une forme infinitive qui suit normalement les mots « ne » et « pas ».	Ne pas fumer.

3.2.4. Désambiguïsation d'homographes partageant la même classe grammaticale

Les indices présentés aux Tableaux 3.13 et 3.14 aident à la désambiguïsation d'homographes dans les cas où ceux-ci n'appartiennent pas à la même classe grammaticale. Ces indices ne sont malheureusement pas utiles, de façon générale, lorsque l'on fait face à des homographes issus de la même classe. Par exemple, utiliser la séquence des mots dans une phrase n'aidera pas à déterminer si le mot « fils » est associé au lemme « fil » (bout de tissu ») ou au lemme « fils » (enfant de sexe masculin). La désambiguïsation de mots appartenant à la même classe grammaticale est donc de façon générale plus complexe.

Toujours est-il que dans certains cas, certains des indices cités plus haut peuvent tout de même s'avérer utiles. Par exemple, pour le mot « fils », le genre et le nombre des déterminants et adjectifs associés à ce nom donnent des indices quant au lemme auquel on a affaire. Par exemple, examinons cette phrase :

« Le fils de mon amie est arrivé ».

Ici, comme le mot « fils » est précédé d'un déterminant au singulier, il est clair qu'il ne peut s'agir de la variation au pluriel du lemme « fil ». Le genre d'un homographe peut aussi parfois être utile. Par exemple, le mot « mousse » est soit féminin (dans le sens de petite plante), ou masculin (dans le sens de jeune marin). Dans un tel cas, le genre du déterminant accompagnant le nom et potentiellement de l'adjectif le cas échéant, peuvent donner un indice du lemme à considérer. Il est parfois à noter que suivant la définition d'homographe adoptée pour ce travail, ces deux variantes de « mousse » ne seraient pas considérées comme des homographes, puisqu'elles partagent la même classe grammaticale, et la même orthographe pour le lemme. Il faut aller plus profondément au niveau de l'étiquetage pour bien faire la distinction entre ces deux formes.

Pour les verbes, certains indices listés plus haut pour des mots de classes grammaticales différentes peuvent aussi s'avérer utiles. Par exemple, pour le mot « suis », examinons l'exemple suivant :

« Tu suis les consignes »

En général, le mot « suis » peut soit être une forme conjuguée du verbe « être » ou du verbe « suivre ». Mais dans le cas du verbe « être », la forme « suis » n'est utilisée qu'à la première personne du singulier. Il est donc clair, dans cet exemple, que le mot « suis » est ici associé au lemme « suivre », car le pronom personnel nominatif « tu » précède le verbe.

On voit donc qu'il y a certains cas où certains indices énoncés précédemment peuvent quand même s'avérer utiles même pour les homographes de même classe grammaticale. Mais il y a évidemment des limites. Considérons cette autre phrase :

« Je suis les consignes »

Cette fois, comme le pronom personnel nominatif précédant le verbe est à la première personne, on ne peut se servir de cet indice pour la désambiguïsation. On pourrait en revanche utiliser comme indice la présence du mot « consignes ». En effet, on retrouve souvent les lemmes « suivre » et « consigne » dans la même phrase. On parle alors de cooccurrences. L'algorithme de désambiguïsation développé pour le présent projet ne considère pas les cooccurrences, car cela en augmenterait la complexité. Il faut se rappeler qu'on ne recherche pas à tout prix une performance de désambiguïsation des homographes très élevée pour ce projet de mémoire.

Ceci dit, les cooccurrences présentes dans le corpus de référence sont comptabilisées au moment de l'analyse du texte, de façon à par la suite bâtir des phrases à l'Étape 2 où on tient compte de la probabilité de retrouver certains mots dans la même phrase. Cela permet de potentiellement bâtir des phrases plus réalistes. Mais surtout, cela permettra, comme on le verra au Chapitre 6,

d'évaluer si des outils de lemmatisation existants peuvent tirer profit de la présence de telles cooccurrences. Les cooccurrences sont discutées davantage à la Section 3.3.

Mais il y a d'autres cas où même l'analyse des cooccurrences ne suffit pas pour la désambigüisation. Considérons la phrase suivante :

> « Je suis ton père ».

Dans un tel cas, il n'est pas possible de déterminer si le verbe « suis » est issu du verbe « être » ou du verbe « suivre ». On ne sait en effet si l'interlocuteur se présente comme le père de celui ou celle à qui il parle, ou s'il indique plutôt qu'il est derrière le père de la personne à qui il parle et qu'il marche derrière le père en question. Pour le déterminer, il faudrait analyser les phrases environnantes d'un point de vue sémantique, ce qui est bien au-delà de la portée du présent projet.

L'algorithme de désambigüisation développé au cours de ce projet ne s'attardera donc pas du tout aux homographes appartenant à la même classe grammaticale. En conséquence, si on retrouve le mot « fils » dans le corpus, les deux lemmes associés seront pris en compte dans le calcul de fréquence des lemmes dans le corpus de référence. Il en résultera qu'à l'Étape 2, même s'il s'avérait que le mot « fils » n'a été utilisé dans le corpus de référence que dans le sens de « bout de tissu », une phrase telle que « Mon fils est arrivé » pourra théoriquement être générée. C'est donc dire que les phrases générées pourraient ainsi s'éloigner quelque peu de ce qu'on retrouve dans le corpus de référence. Mais cette conséquence n'est pas très fâcheuse, car encore faut-il le rappeler, la désambiguïsation des homographes n'est pas l'objectif principal de ce projet de recherche.

3.2.5. Identification des syntagmes pour faciliter la désambiguïsation

Selon le dictionnaire en ligne Usito (2024), un syntagme est un « groupe de mots formant une unité dans une organisation hiérarchisée de la phrase ». De telles « unités » sont donc composées de mots qui se retrouvent fréquemment groupés, qu'on peut décrire comme étant des « cooccurrences » ou « collocations ». Les syntagmes se distinguent par leur degré de cohésion. Par exemple, dans un syntagme dit « figé », comme le décrit Larivière (1998), « le degré de figement est total, puisque aucun élément ne peut s'insérer entre les éléments figés ». On parle alors aussi de locution. Larivière ajoute : « le sens individuel de chacun des éléments est perdu au profit du sens du tout au point où il n'y a pas de commutation possible des éléments ». On peut citer en exemple certaines locutions ou mots composés classifiés comme syntagmes figés:

- Donner un chèque en blanc.
- « cordon bleu »
- « canapé-lit »

De tels syntagmes figés sont particulièrement utiles dans le contexte de la désambiguïsation. En effet, lorsqu'identifiés dans un texte, ceux-ci nous permettent de facilement définir la classe grammaticale de chacun des mots qui les composent.

Prenons comme exemple le syntagme figé « pomme de terre ». Ce syntagme comporte trois homographes, « pomme », « de » et « terre ». En effet, le mot « de » peut être soit un déterminant, soit une préposition. De la même façon, les mots « pomme » et « terre » peuvent être soit un nom, soit une forme conjuguée des verbes « pommer » ou « terrer ». La désambiguïsation de ce syntagme peut s'effectuer sur la base des règles de grammaire identifiées à la Section 3.2.3. Mais le processus de désambiguïsation est grandement facilité lorsque ces mots sont d'emblée reconnus comme composant un syntagme figé. Dans le cas précis de « pomme de terre », on sait alors que « de » est une préposition et que « pomme » et « terre » sont des noms communs, sans avoir à se soucier de l'environnement dans lequel se retrouve le

syntagme. Il est théoriquement possible que les trois mots « pomme », « de » et « terre » se retrouvent l'un à la suite de l'autre dans un contexte autre que celui du légume, mais la probabilité que cela se produise est très faible. Il est intéressant de noter que Usito (2024) comprend comme entrée le syntagme « pomme de terre » considéré dans son tout comme un nom féminin. Les concepteurs d'un lemmatiseur doivent donc décider comment doivent être traités de tels syntagmes figés : soit comme un tout, soit comme un ensemble de mots individuels. Mais dans un cas comme dans l'autre, le repérage de syntagmes figés dans un texte constitue donc une première étape stratégique d'un processus de désambiguïsation. En effet, Bourdaillet et Ganascia (2005) confirment que l'algorithme pour leur étiqueteur prend en compte les locutions, en cherchant dans leur base de données du lexique de base, la plus grande locution correspondant à une série de mots du texte.

L'algorithme de désambiguïsation développé pour ce projet se voulant simple, les seuls syntagmes qui y ont été considérés sont ceux facilitant la désambiguïsation d'homographes pouvant être à la fois des verbes et des noms communs. Quelques exemples sont fournis ici :

- « faire place », « prendre place »
- « livrer bataille »
- « prendre garde »
- « tenir compte », « rendre compte »
- « perdre contrôle », « prendre contrôle »
- « faire part », « prendre part »
- « faire partie »
- « prendre note »
- « faire gaffe »
- « rendre visite »
- « faire semblant »
- « faire mine »
- « faire signe »
- « avoir envie », « faire envie »
- « rendre grâce », « trouver grâce », « faire grâce »

On constate en premier lieu que tous les syntagmes cités plus haut permettent de déterminer que l'homographe situé en deuxième position est un nom commun, et non une forme verbale. Ces syntagmes ne sont pas « figés », puisqu'ils impliquent des verbes pouvant être conjugués à n'importe quel temps ou personne (« tu prenais place », « le policier lui fera signe »), et que certains mots tels que des adjectifs ou adverbes, peuvent dans certains cas s'y intercaler (« je prends bien note de ce détail »).

Aucun effort particulier ne sera mis pour introduire de tels syntagmes dans les phrases générées aléatoirement à l'Étape 2 de ce projet, mais il y a tout de même un certain potentiel de les y retrouver. En effet, puisqu'on considère à l'Étape 2 les cooccurrences de chaque mot (tel qu'on le discutera à la Section 3.3), il est possible que certains de ces syntagmes réapparaissent dans les phrases générées au hasard.

3.2.6. Utilisation de statistiques globales du corpus pour la désambiguïsation

Tous les indices présentés aux sections précédentes pour la désambigüisation sont en lien avec uniquement les mots de la phrase courante sous analyse. En poussant plus loin l'analyse et en considérant le corpus dans son entièreté, on peut retirer d'autres indices pertinents pour la désambiguïsation des homographes. En effet, une analyse statistique peut aider à déterminer lequel de deux homographes a le plus de chances d'être le plus approprié dans la phrase étudiée.

Par exemple, le mot « plus » peut soit être un adverbe (« je ne veux *plus* jouer avec toi ») ou le passé simple du verbe « plaire » à la première ou la deuxième personne du singulier (« je *plus*, tu *plus* »). Devant un tel homographe, on peut évidemment se servir des indices développés aux Tableaux 3.13 et 3.14, puisqu'il est ici question d'homographes de deux classes grammaticales différentes (adverbe ou verbe). Mais on peut aussi récolter d'autres indices, en observant certaines statistiques se rapportant au corpus en entier. On pourrait par exemple analyser la fréquence d'utilisation du passé simple dans le corpus. Si on remarque qu'outre les formes ambiguës reliées aux homographes, le passé simple n'est pas ou très peu utilisé, il est fort à parier que le mot « plus » rencontré correspond à l'adverbe plutôt qu'au verbe conjugué. On pourrait aussi s'attarder spécifiquement à la présence du verbe « plaire » dans le corpus. Si on constate que ce verbe n'est jamais ou très peu utilisé dans les formes autres que « plus », il est aussi peu probable que les occurrences du mot « plus » dans le corpus correspondent à la forme verbale.

Il s'agit encore là uniquement d'indices. Il est en effet théoriquement possible que le verbe « plaire » ne soit employé qu'une seule fois, et au passé simple, dans un texte donné, malgré le fait que le passé simple soit autrement très rare dans le texte. Mais de tels indices statistiques, ajoutés aux indices des Tableaux 3.13 et 3.14, peuvent influer positivement sur le succès potentiel de l'analyse des homographes. Dans le cadre de ce mémoire, le portrait statistique du corpus de référence en entier n'a été étudié que pour désambiguïser les formes verbales, qu'on doit distinguer d'autres classes grammaticales, comme dans l'exemple de l'homographe « plus » cité plus haut. Comme autres exemples, on peut citer les homographes « demande », « contrôle », etc.

Deux algorithmes statistiques ont été mis en place pour l'outil actuel. Pour le premier test, l'algorithme vérifie si le verbe en cause ne se retrouve que sous deux formes conjuguées dans le texte, et que ces deux formes correspondent aussi à des noms ou adjectifs dont l'une est le pluriel de l'autre. Prenons par exemple l'homographe « dame », qui peut être soit une forme conjuguée du verbe « damer », ou le nom commun « dame ». Si l'algorithme ne localise par exemple dans l'ensemble du corpus que deux formes verbales potentielles du verbe damer, soit « dame » et « dames », il est fort à parier qu'on fait affaire ici avec le nom commun « dame ». En effet, les mots « dame » et « dames » sont issus du même lemme et un mot est le pluriel de l'autre. Si l'on avait retrouvé dans le texte d'autres formes conjuguées du verbe « damer », telles que « damais », « damerai », ou même l'infinitif « damer », la probabilité que l'homographe « dame » ait été un verbe aurait été déterminée comme étant plus élevée.

Pour le deuxième test, on assume que l'homographe est une forme verbale et on calcule le ratio des occurrences de la forme verbale courante (temps et personne) sur l'ensemble de toutes les formes du même verbe. On compare ensuite ce ratio avec ce même ratio (temps et personne courants vs. toutes les formes verbales) en tenant compte de tous les verbes, excluant le verbe courant. Reprenons par exemple l'homographe « plus » discuté précédemment. Si celui-ci est un verbe, il est le passé simple du verbe « plaire » conjugué à la première ou la deuxième personne du singulier. On calcule donc le ratio des apparitions dans le corpus des deuxième et troisième personnes du singulier du passé simple du verbe « plaire » sur l'ensemble des apparitions du verbe « plaire » à toutes les personnes et à tous les temps. Une valeur élevée de ce ratio pour le verbe « plaire », en comparaison avec ce même ratio calculé sur la base de tous les verbes

présents dans le corpus de référence, donne un indice comme quoi le mot « plus » n'est pas une forme verbale, mais plutôt un adverbe dans le cas présent.

Il faut rappeler que les deux tests statistiques présentés ici nous donnent des *indices* comme quoi un homographe donné est ou n'est pas un verbe. Il n'est pas question ici de détermination absolue. En conséquence, les algorithmes statistiques présentés plus en détail au Chapitre 4 modifieront les probabilités d'appartenance à une classe grammaticale, sans pour autant les classifier de façon certaine. Le succès de ces algorithmes sera discuté au Chapitre 5.

Plutôt que de se fier uniquement sur l'analyse du corpus de référence choisi comme on le propose ici, on pourrait plutôt utiliser des statistiques observées sur de plus vastes corpus de langue française analysés par d'autres chercheurs. De plus, il n'est pas nécessaire de se limiter qu'aux homographes comportant au moins une forme verbale comme on l'a proposé plus haut. Par exemple, Vergne et Giguet (1998), suite à l'analyse d'un large corpus lemmatisé manuellement, ont observé que seuls 2,4% des homographes « le », « l' », « la » et « les » se réfèrent à des pronoms, plutôt qu'à des déterminants[3]. Ces auteurs suggèrent ainsi qu'une désambiguïsation dite « par défaut » puisse être appliquée, en assignant automatiquement la classe grammaticale majoritaire à de tels homographes. On s'attend alors à un taux d'erreur de l'ordre d'aussi peu que 2,4% pour ce cas précis. Même dans le cas où on n'appliquerait pas d'étiquetage par défaut, ces statistiques peuvent tout de même fournir des indices additionnels au processus de désambiguïsation.

3.3. Identification des cooccurrences

L'objectif premier du présent projet est de générer de façon aléatoire des textes automatiquement lemmatisés. Peu d'efforts sont déployés pour que ces textes aléatoires soient dotés de sens, car l'évaluation d'outils de lemmatisation ne dépend généralement pas de la sémantique. Toujours est-il que dans l'espoir de donner un minimum de sens aux phrases aléatoires, une analyse des cooccurrences (ou collocations) est effectuée sur le corpus de référence choisi. En effet, en repérant les mots qu'on retrouve fréquemment autour d'un mot donné au sein du corpus de référence, il devient ensuite possible de faire apparaître à nouveau ces cooccurrences autour du même mot, lors de la génération de phrases au hasard. À l'outil de lemmatisation créé à l'Étape 1, un algorithme a donc été ajouté pour identifier les cooccurrences des mots appartenant à certaines classes grammaticales, soit les verbes, adjectifs, noms communs et adverbes. Ces classes ont été choisies sur la base que les mots qui les composent sont ceux contenant le plus de valeur sémantique.

Ces cooccurrences, autres mots retrouvés dans l'environnement immédiat d'un mot étudié, seront ensuite utilisées pour bâtir à l'Étape 2 des textes comportant possiblement un *minimum* de sens. Par exemple, une fois un verbe choisi aléatoirement, l'algorithme puise en priorité dans la liste de cooccurrences de ce même verbe observée dans le corpus de référence, pour sélectionner les autres mots composant la phrase à créer.

Au-delà de la recherche de sens, l'incorporation de telles cooccurrences offre aussi des indices potentiels à des lemmatiseurs quant à la fonction grammaticale de certains mots. Cela est particulièrement le cas quand la fonction syntaxique et la structure de la phrase seules ne suffisent pas à assurer une désambiguïsation certaine.

Pour illustrer la méthode adoptée pour le repérage de cooccurrences, utilisons deux phrases quelconques à titre d'exemple :

[3] Comme on le verra au Chapitre 5, l'analyse des corpus de référence considérés pour ce projet indique plutôt que les pronoms représentent 12,4% des cas pour ces homographes.

Phrase 1 : Les voitures rouges roulent rapidement.

Phrase 2 : Ces voitures passent sur la lumière rouge.

Pour l'analyse des cooccurrences, tel que mentionné plus haut, nous ne nous intéressons ici qu'à quatre classes de mots : les verbes, les noms communs, les adjectifs et les adverbes. Dans ces deux phrases, nous allons donc extraire uniquement les mots appartenant à ces quatre classes grammaticales. Pour faciliter l'illustration, on utilise un code de couleurs. Un fond vert est utilisé pour identifier les verbes, un fond rouge pour les adjectifs, bleu pour les noms et finalement jaune pour les adverbes. On réécrit donc ces deux phrases en utilisant ce code de couleurs :

Phrase 1 : Les voitures rouges roulent rapidement.

Phrase 2 : Ces voitures passent sur la lumière rouge.

Comme l'analyse présente ne se penche que sur les mots mis en évidence avec ces couleurs, les mots appartenant aux autres classes grammaticales (déterminants, pronoms, etc.) peuvent être omis. On se retrouve alors avec :

Phrase 1 : voitures rouges roulent rapidement

Phrase 2 : voitures passent lumière rouge

Il est plus approprié d'effectuer l'analyse des cooccurrences sur la base des lemmes des mots, plutôt que sur leurs formes fléchies. En effet, que l'on parle de « voitures rouges » ou de « voiture rouge » (singulier ou pluriel), il est toujours question d'associer deux concepts précis, soit celui de « voiture » et celui de la couleur rouge. On remplace donc les formes fléchies, le cas échéant, par leurs formes de base ou canoniques, donc leurs lemmes :

Phrase 1 : voiture rouge rouler rapidement

Phrase 2 : voiture passer lumière rouge

La méthode adoptée ici ne recherche les cooccurrences qu'à l'intérieur d'une phrase donnée. L'algorithme aurait aussi pu puiser dans les phrases environnantes, mais par simplicité, on se limite aux mots situés au sein d'une seule phrase. Ces cooccurrences sont nécessairement ainsi les plus « fortes ». Pour chaque lemme, on compile ses cooccurrences selon les quatre classes grammaticales identifiées, créant ainsi quatre sous-ensembles pour chaque lemme, un sous-ensemble correspondant à chacune des classes.

Ainsi, pour le lemme « voiture » dans la première phrase, on retrouve trois cooccurrences : « rouge », « rouler » et « rapidement ». Celles-ci appartiennent à des classes grammaticales différentes. Pour le lemme « voiture », on liste donc « rouge » dans la catégorie « adjectifs », « rouler » dans la catégorie « verbes », et rapidement dans la catégorie « adverbes ». Ce résultat est illustré au Tableau 3.15. On procède ainsi de la même façon pour les trois autres mots d'intérêt de la Phrase 1. Par exemple, pour le lemme « rouge », nous obtenons le résultat fourni au Tableau 3.16.

On analyse ensuite la Phrase 2. On y retrouve une fois de plus le lemme « voiture ». On ajoutera donc, pour le lemme « voiture », les nouvelles cooccurrences trouvées cette fois dans la deuxième phrase. On aboutit donc au résultat illustré au Tableau 3.17.

Tableau 3.15 : Cooccurrences pour le lemme « voiture » - première phrase

Verbes	Noms communs	Adjectifs	Adverbes
rouler		rouge	rapidement

Tableau 3.16 : Cooccurrences pour le lemme « rouge » - première phrase

Verbes	Noms communs	Adjectifs	Adverbes
rouler	voiture		rapidement

Tableau 3.17 : Cooccurrences pour le lemme « voiture » - deux phrases

Verbes	Noms communs	Adjectifs	Adverbes
rouler passer	lumière	rouge (2)	rapidement

On constate au Tableau 3.17 que deux verbes sont maintenant associés au lemme « voiture », soit « rouler » et « passer ». De plus, on constate que bien que l'adjectif « rouge » demeure pour l'instant le seul adjectif associé à « voiture », on en retrouve désormais deux occurrences (une dans chaque phrase). Cette information est illustrée ici entre parenthèses, ce qui permet de comptabiliser la fréquence d'apparition de chaque cooccurrence.

Une fois de plus, on procède ainsi pour chaque autre lemme des deux phrases, et le processus se répète ensuite pour toutes les autres phrases du corpus de référence. On se retrouve donc en bout de ligne avec autant de tableaux qu'on retrouve de lemmes distincts de verbes, noms communs, d'adjectifs et d'adverbes dans le corpus de référence. Et ces tableaux sont toujours composés de quatre colonnes, chacune correspondant à une des quatre classes grammaticales visées. Certaines colonnes peuvent se retrouver vides.

Ces tableaux de cooccurrences sont donc bâtis à l'Étape 1, au moment de l'analyse du corpus de référence. Ils ne sont cependant utilisés que plus tard, à l'Étape 2, quand vient le temps de générer des phrases aléatoires. L'utilisation de ces tableaux de cooccurrences pour l'Étape 2 est décrite à la Section 4.4.

Il faut toutefois distinguer le concept de cooccurrence adopté pour ce projet à celui plus précis de collocation décrit par Larivière (1998). Celle-ci recherche des syntagmes, figés ou non, présents dans un texte. Ces syntagmes consistent en des mots regroupés en unités, donc apparaissant l'un à la suite de l'autre et dans un ordre généralement déterminé. Pour ce projet, on ne se soucie ni du fait que les cooccurrences se suivent directement, ni de l'ordre dans lequel elles apparaissent. On ne se soucie en effet que de leur présence commune dans une phrase donnée. Il n'est donc pas ici question de syntagmes, locutions ou collocations, tels que discutés à la Section 3.2.5.

3.4. Sélection des corpus de référence

Aux sections précédentes du Chapitre 3, il a été question de l'utilisation d'un corpus de référence. Le corpus de référence joue ici deux rôles. En premier lieu, il nous sert de source de données pour les mots qui seront utilisés à l'Étape 2 pour la génération de textes aléatoires automatiquement lemmatisés. En effet, il aurait été trop ardu, en particulier dans le contexte d'un projet de mémoire s'étalant sur un échéancier relativement limité, de considérer une liste quasi exhaustive de tous les mots du dictionnaire. D'autant plus que même les grands dictionnaires

« grand public » courants ne renferment en fait qu'une partie du lexique total du français. Une approche moins ambitieuse a donc été adoptée pour les besoins de ce projet. On y a choisi deux textes libres d'accès, dont on a extrait une liste de noms, adjectifs, verbes et adverbes. Le premier de ces textes est le roman français « Le Rouge et le Noir » de Stendhal. Ce roman, depuis longtemps libre de droits (ce texte date de 1831) porte sur la vie villageoise de la France du 19e siècle. Il comporte plus de 180 000 mots, une longueur appropriée pour l'analyse présente. Le deuxième texte est un roman de science-fiction non publié, écrit par l'auteur de ce mémoire. Ce second roman a l'avantage d'être forcément entièrement accessible et libre de droits, en plus d'être lui aussi d'une longueur totale appropriée pour l'étude en cours (près de 200 000 mots). De plus, la familiarité avec son contenu a facilité la mise au point des algorithmes, en particulier au niveau du débogage informatique. Des extraits de ces deux romans sont fournis à l'Annexe A en référence.

Ce sont donc ces deux romans qui ont servi de corpus de référence, et qui ont été soumis à l'algorithme de lemmatisation de base développé pour ce projet. L'outil de lemmatisation en a donc extrait les lemmes des quatre principales classes grammaticales, soit les verbes, les noms communs, les adjectifs et les adverbes. Ce ne sont que ces lemmes présents dans le corpus de référence, et non l'ensemble des lemmes connus de la langue française, qui sont par la suite utilisés à l'Étape 2 du projet pour la génération aléatoire de phrases.

Un autre rôle potentiellement important joué par le corpus de référence est de positionner le texte à générer au hasard dans un domaine précis du domaine écrit. On peut en effet ainsi choisir un type de texte en particulier, appartenant à un champ de connaissances en particulier, par exemple, la littérature, la finance, ou la médecine. Comme le mentionne Grouin (2022), le choix du corpus permet ainsi de s'adapter au besoin, à un registre de la langue en particulier (académique, familier, soutenu, etc.) ou à des usages particuliers, tels que les courriers électroniques, les réseaux sociaux, etc. En effet, comme le précise Grouin, de tels documents peuvent engendrer « des pratiques linguistiques particulières aux niveaux orthographique, sémantique et syntaxique, qui peuvent s'apparenter à des mots inconnus de la langue générale ». Il ajoute qu'il a été démontré que « l'analyse de ces productions langagières au moyen d'outils de traitement automatique des langues est de meilleure qualité si les ressources développées (lexiques, modèles statistiques) sont représentatives des usages qu'elles doivent traiter », citant au passage L'Homme (2008).

Donc, on peut s'attendre pour ce projet, vu le corpus choisi, que les phrases générées aléatoirement à l'Étape 2, fassent appel à un lexique relié soit à la réalité villageoise de la France du 19e siècle si on se base sur « Le Rouge et le Noir » de Stendhal, soit à un univers de science-fiction si on se base sur le deuxième corpus de référence choisi. Alternativement, la sélection d'un corpus de référence issu par exemple de la biochimie, aurait de la même façon fait générer des phrases à l'Étape 2 comportant bon nombre de termes techniques en lien avec la biochimie, avec des cooccurrences inspirées de ce domaine. Donc, bien que les corpus de référence choisis ici soient spécifiques, l'algorithme en place demeure général et permet donc d'ajuster les phrases générées aléatoirement en fonction d'un domaine de connaissances précis.

3.5. Étape 2 : Génération de textes aléatoires automatiquement lemmatisés

A l'Étape 2, on cherche maintenant à générer un nombre quelconque de phrases aléatoires automatiquement lemmatisées. Autrement dit, ces phrases ne sont pas lemmatisées « après coup », mais au contraire au moment même de leur création, en suivant des modèles de phrases prédéterminés. L'objectif visé est d'ultimement fournir des textes servant à l'évaluation d'outils de lemmatisation existants en les confrontant à des textes déjà lemmatisés servant d'étalon doré (Chapitre 6). La pertinence de cet outil repose sur le fait que la lemmatisation manuelle d'un texte, surtout s'il est volumineux, représente un exercice très fastidieux. De façon générale, afin d'éviter

d'avoir eux-mêmes à effectuer cette tâche, les linguistes utilisent de larges corpus de référence ayant été lemmatisés manuellement par d'autres équipes de chercheurs. Par exemple, la banque FRANTEXT, telle que décrite par Bernard et al. (2002), leur est disponible. Celle-ci est composée d'un vaste corpus de textes composés à 80% d'œuvres littéraires et de 20% d'œuvres scientifiques ou techniques, dont une grande partie a été soumise à un codage grammatical manuel. Hug (2002), cité plus haut, a fait usage de cette banque de textes. Bourdaillet et Ganascia (2005) quant à eux, ont utilisé le corpus GRACE, lui aussi annoté morpho-syntaxiquement, comportant près de 800 000 mots. Citant Elworthy (1994), ces auteurs mentionnent d'ailleurs que l'apprentissage d'un modèle d'étiquetage basé sur un corpus manuellement étiqueté produit de meilleurs résultats qu'une procédure d'apprentissage basée uniquement sur des algorithmes.

L'utilisation commune de corpus déjà existants permet aux chercheurs de comparer l'efficacité de leurs outils sur la base des mêmes textes de référence. Cependant, l'utilisation répétée des mêmes corpus limite ultimement la portée des recherches, car on ne s'expose pas nécessairement à des textes représentatifs d'autres domaines de connaissances, d'autres niveaux de langage ou reliés à d'autres contextes, comme on le mentionnait à la Section 3.4. Les textes aléatoires générés au cours du présent projet cherchent justement à palier cette faiblesse.

L'Étape 2, dont le but est de générer ces textes automatiquement lemmatisés, représente donc le cœur de ce projet, car l'Étape 1 n'aura servi qu'à fournir le matériel de base nécessaire pour effectuer l'Étape 2. À cette étape, on définit d'abord un certain nombre de structures de phrases typiques. Une analyse même rapide de tout texte en français nous révèle la grande variété et complexité de phrases qu'on peut y retrouver. L'outil développé ici, surtout dans le contexte d'un mémoire de maîtrise limité dans sa portée et n'impliquant qu'un seul individu, ne peut considérer qu'un nombre restreint de structures de phrases définies par l'agencement et la séquence de mots appartenant à différentes classes grammaticales.

En sortie de l'Étape 2, on dispose finalement, en plus du texte aléatoire lui-même, de banques de données contenant les caractéristiques de chacun des mots du texte. Ces caractéristiques sont le lemme, la classe grammaticale, ainsi qu'un étiquetage morpho-syntaxique plus précis. De plus, plusieurs statistiques globales sont comptabilisées, permettant une comparaison globale de la performance d'outils de lemmatisation à évaluer, au Chapitre 6.

3.5.1. Information requise de la lemmatisation du corpus de référence

Pour générer des phrases aléatoires automatiquement lemmatisées, l'algorithme développé à l'Étape 2 doit piger dans des banques de données de mots de différentes classes grammaticales. Tel que mentionné plus haut, plutôt que d'utiliser le lexique complet de la langue française, l'algorithme puise dans le corpus de référence préalablement lemmatisé à l'Étape 1. Il en extrait les lemmes de tous les mots du corpus appartenant aux quatre classes grammaticales suivantes :

- Verbes
- Noms communs
- Adjectifs
- Adverbes

C'est donc parmi ces lemmes et uniquement parmi ceux-ci, que les verbes, noms communs, adjectifs et adverbes sont choisis pour générer les phrases aléatoires de l'Étape 2. Pour les autres classes grammaticales, comme les déterminants et pronoms, des banques génériques passablement exhaustives sont utilisées. En sélectionnant les lemmes parmi ceux apparaissant dans le corpus de référence, on s'assure que les phrases générées aléatoirement soient représentatives du corpus, d'un point de vue lexical.

De plus, l'algorithme de lemmatisation de l'Étape 1 compile aussi la proportion approximative des temps de verbe utilisés, afin de guider la sélection des temps de verbes pour les phrases de l'Étape 2. L'algorithme de l'Étape 2 offre toutefois la possibilité de modifier la proportion de ces temps de verbes au besoin, afin de s'assurer par exemple que tous les temps de verbe soient utilisés, ou pour favoriser pour une raison ou une autre, certains temps en particulier. Il en va de même pour les personnes des verbes, qui sont aussi comptabilisées à l'Étape 1. Encore là, l'algorithme de l'Étape 2 offre la possibilité d'ajuster ces fréquences manuellement.

L'information de l'Étape 1 requise pour procéder à l'Étape 2 peut donc se résumer comme suit :

- Liste de tous les lemmes pour les verbes, noms communs, adjectifs et adverbes
- Fréquence d'apparition absolue de chacun de ces lemmes dans le corpus de référence
- Fréquence des temps de verbes (e.g. présent, imparfait, etc.)
- Fréquence des personnes de verbes (premières, deuxièmes et troisièmes personnes du singulier et du pluriel)

Mais ces listes et fréquences sont comptabilisées de trois façons :

1. Sur la base des mots non ambigus uniquement
2. Sur la base de toutes les possibilités d'homographes
3. À la suite d'une désambiguïsation complète

Dans le premier cas, on exclut les homographes de l'analyse. C'est donc dire qu'à chaque fois que plus d'un lemme ou plus d'une classe grammaticale est possible pour un mot du corpus, on ne comptabilise pas ce mot dans les statistiques du corpus. Le but de ce procédé est de s'assurer qu'aucun mot ou lemme ne faisant pas réellement partie du corpus se retrouve utilisé à l'Étape 2. Par exemple, si le mot « dame » apparaît, il ne sera pas comptabilisé, car il peut signifier soit le nom commun « dame », soit une forme conjuguée du verbe « damer ». On évite ainsi, par exemple, que le verbe « damer » soit utilisé pour former des phrases, dans un cas où ce verbe ne serait jamais utilisé dans le corpus de référence. Le désavantage de cette approche est que certains mots possiblement très fréquents au sein du corpus de référence se retrouvent à être exclus, ce qui fait en sorte que les phrases de l'Étape 2 ne sont pas aussi représentatives du corpus qu'on le voudrait.

Dans le deuxième cas, on inclut toutes les possibilités d'homographes. C'est donc dire qu'à chaque fois que l'algorithme croise un homographe, il considère en parallèle toutes ses possibilités homographiques. Ainsi, si l'algorithme croise le mot « dame », les fréquences d'apparition du nom commun « dame » et du verbe « damer » sont toutes deux mises à jour, peu importe la réelle classe grammaticale du mot dans le contexte de la phrase dont il est extrait. Il en résulte que le nombre de lemmes comptabilisés de cette façon est substantiellement plus grand que pour la première approche.

Le résultat de la troisième approche, basée sur une désambiguïsation complète, est la plus intéressante. Avec cette approche, en supposant une désambiguïsation parfaite, on s'assure d'une part que tous les lemmes du corpus sont disponibles pour l'Étape 2, contrairement à ce qu'on obtient avec la première approche. On s'assure aussi que seuls les lemmes apparaissant réellement dans le corpus de référence sont utilisés à l'Étape 2, contrairement à la deuxième approche. Comme l'outil de lemmatisation développé ici n'est pas parfait, il en découle que certains lemmes non présents dans le corpus risquent de se glisser dans la banque de lemmes disponibles pour l'Étape 2. Mais plus la performance de désambiguïsation augmente, plus la présence de tels mots « imposteurs » est réduite. Les particularités des trois approches correspondant aux trois banques de données mentionnées plus haut sont illustrées au Tableau 3.18.

Tableau 3.18 : Particularités des trois approches pour l'extraction des lemmes

Approche	Tous les lemmes du corpus de référence sont inclus	Aucun lemme absent du corpus de référence n'est inclus
1 – Formes non ambiguës seulement	✗	✓
2 – Tous les homographes considérés	✓	✗
3 – À la suite d'une désambiguïsation complète et parfaite	✓	✓

On déduit du Tableau 3.18 que l'approche idéale à adopter est la troisième, si l'on cherche à générer des phrases le plus représentatives possible du corpus de référence. Cela confirme la pertinence de la désambiguïsation du corpus décrite à la Section 3.2.1. Cependant, on n'obtient un résultat optimal que si le processus de désambiguïsation est parfait, ce qui est peu probable, surtout considérant qu'un algorithme relativement simple a été mis en place à l'Étape 1 pour ce projet. À tout le moins, même imparfaite, la désambiguïsation favorise la création de phrases aléatoires plus représentatives du corpus de référence que si l'on adopte l'une des deux autres approches suggérées au Tableau 3.18. Au Chapitre 5, on comparera des résultats obtenus selon ces trois approches.

Finalement, d'autres informations sont aussi extraites du corpus à l'Étape 1, mais à titre de référence uniquement, ne servant pas directement à la génération de phrases pour l'Étape 2 :

- Le nombre de mots du corpus
- Le nombre de phrases du corpus
- La longueur moyenne des phrases du corpus
- La liste complète d'homographes et leurs classes grammaticales associées

Ces informations et les statistiques y étant rattachées sont uniquement comparées avec les résultats de l'Étape 2, une fois les phrases aléatoires générées, pour augmenter la portée de la discussion et de l'analyse.

3.5.2. Structures des phrases à générer

Les phrases servent à exprimer à l'oral ou à l'écrit un flot de pensées, en agençant dans un ordre linéaire les différents mots servant à transposer ces pensées. Plusieurs structures de phrases différentes peuvent servir à exprimer la même idée, en interchangeant l'ordre des composantes. Les phrases suivantes, citées en exemple par Yvon (2010), expriment une idée commune :

« Jean casse la boîte. »

« La boîte est cassée par Jean. »

« Jean, il casse la boîte. »

« La boîte, c'est Jean qui la casse ».

On pourrait aussi inclure davantage d'éléments à une phase, et modifier la séquence dans laquelle ces éléments apparaissent, comme dans ces exemples :

« Hier, je suis allé au marché avec un ami. »

« Je suis allé au marché hier avec un ami. »

« Je suis allé hier au marché avec un ami. »

Encore plus que dans la série d'exemples précédente impliquant « Jean et la boîte », on a affaire ici avec exactement le même sens pour ces trois phrases. Seule l'emphase change. Par exemple, dans la première phrase, on met davantage l'emphase sur le fait que l'action s'est produite hier.

On constate donc qu'on peut exprimer une même idée avec une multitude de structures de phrases différentes. La langue française, comme possiblement toutes les autres langues, offre en effet un large éventail de structures possibles, permettant de s'adapter aux besoins du locuteur. Ces structures peuvent être exprimées sous la forme d'un « arbre syntaxique ». Comme le précise Yvon (2010), un tel arbre « permet d'identifier simultanément les frontières de constituants, ainsi que les relations de dominance qu'ils entretiennent ». Comme le précise Pinker (2015), il est utile de visualiser l'ordre des mots dans une phrase en les illustrant aux extrémités des branches d'un arbre inversé, pour mieux comprendre comment fonctionne la syntaxe. Pinker (2015) fournit d'ailleurs plusieurs exemples où les mêmes mots, combinés de façons différentes selon divers arbres syntaxiques, peuvent mener à des interprétations divergentes et parfois loufoques de ce même ensemble de mots. La Figure 3.4 donne un exemple d'arbre syntaxique.

Pour le besoin de ce projet, un arbre syntaxique simplifié s'écartant quelque peu du modèle traditionnel est adopté. On ne considère en effet que trois groupes dans la phrase, toujours dans le même ordre. Les phrases générées à l'Étape 2 emploient en effet la structure suivante : groupe sujet – groupe verbe – groupe complément. Pour ce travail, le « groupe verbe » ne comprend que le verbe lui-même, ainsi que possiblement des adverbes s'y rattachant ou des pronoms dans le cas de verbes pronominaux. Le « groupe complément » comprend autant les compléments d'objet direct qu'indirect, auxquels des compléments du nom peuvent se jouxter. Cette approche facilite la programmation des algorithmes. La phrase illustrée à la Figure 3.4, dans le contexte de ce travail de recherche, prend donc plutôt la forme illustrée à la Figure 3.5.

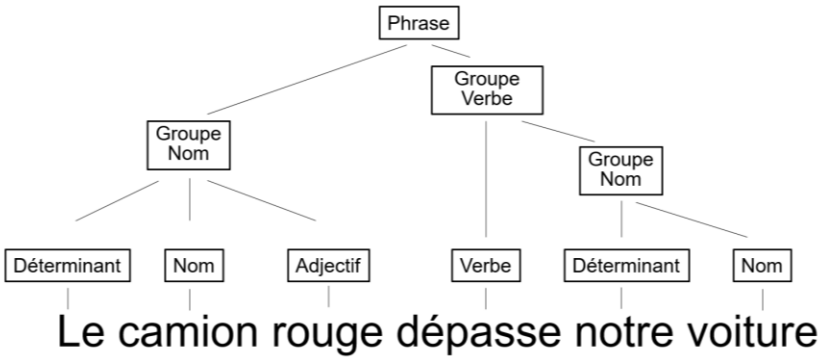


Figure 3.4 : Exemple d'arbre syntaxique traditionnel

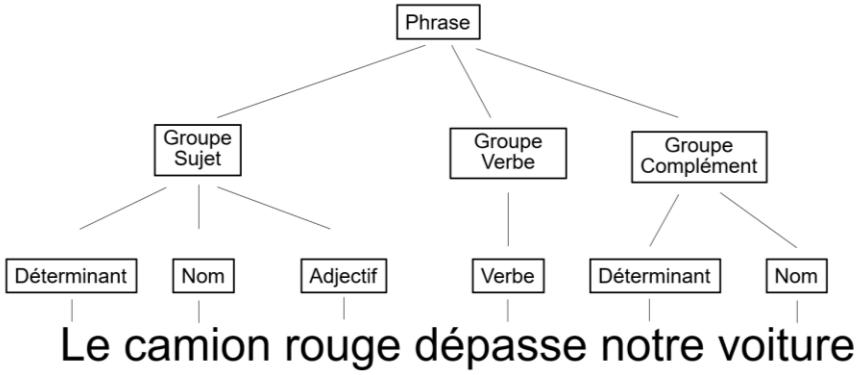


Figure 3.5 : Approche adoptée d'arbre syntaxique et de nomenclature des groupes

La Section 3.5.3 décrit comment ces trois grands groupes (sujet, verbe et complément) sont créés dans le contexte de ce travail de recherche.

3.5.3. Méthodes pour générer les différents groupes de la phrase

L'algorithme de génération automatique de textes de l'Étape 2 crée les phrases les unes après les autres, en suivant le modèle « groupe sujet – groupe verbe – groupe complément » introduit à la Section 3.5.2, jusqu'à ce que le document atteigne le nombre de phrases déterminé par l'utilisateur. Comme il forme le cœur de la phrase, c'est le « groupe verbe » qui est créé en tout premier lieu. Par la suite, le « groupe sujet » est généré, en s'assurant de l'accord du sujet avec les paramètres (personne, genre, nombre) du « groupe verbe ». Finalement, le « groupe complément » est créé, encore là en s'assurant un accord avec les paramètres du « groupe verbe » et aussi parfois avec le groupe du sujet. Par exemple, certains verbes demandent un complément d'objet direct, d'autres demandent un complément d'objet indirect à l'aide d'une préposition précise.

Bien que toutes les phrases générées aléatoirement pour ce projet contiennent un « groupe verbe », certaines ne contiennent aucun « groupe sujet » ou aucun « groupe complément ». En effet, un verbe à l'impératif ne nécessite aucun sujet. Et l'emploi de verbes intransitifs n'implique aucun complément. Les cas généraux, de même que ces cas particuliers, sont discutés aux Sections 3.5.3.1 à 3.5.3.3.

3.5.3.1. Génération du groupe du verbe

Comme le groupe du verbe est central à toutes les phrases, c'est lui qu'on détermine en premier lieu pour ce projet. Un infinitif est tout d'abord sélectionné aléatoirement, parmi tous les verbes figurant dans le corpus de référence, et non pas parmi la liste de verbes plus exhaustive extraite du Bescherelle. Un poids est associé à chaque infinitif, en fonction de sa fréquence dans le corpus de référence. Ainsi, un verbe qui est fréquent dans le corpus le sera aussi dans les textes générés aléatoirement. Ensuite, on détermine, au hasard aussi, le temps du verbe, parmi les 20 possibilités fournies au Tableau 3.4. Encore là, un poids est assigné à chaque temps, en fonction de sa fréquence dans le corpus de référence. Toutefois, l'algorithme permet optionnellement par exemple d'assigner un poids minimum à chaque temps de verbe, s'il est considéré important que tous les temps soient représentés au sein des phrases générées au hasard à l'Étape 2.

Certains temps de verbes du Tableau 3.4 sont composés. Les temps composés sont ceux associés aux codes 12 et plus dans ce tableau, comme le passé composé, le plus-que-parfait et le futur antérieur. Aux temps composés, les verbes doivent être accompagnés de l'auxiliaire

« être » ou de l'auxiliaire « avoir ». Dans le tableau complet des verbes à l'infinitif issu du Bescherelle, à chaque verbe est associé son auxiliaire. La majorité des verbes se conjuguent aux temps composés avec l'auxiliaire « avoir », et certains peuvent se conjuguer avec « être » ou « avoir », comme « passer », « sortir » ou « descendre ». Cependant, quand ces verbes permettant normalement l'usage de l'un ou l'autre des auxiliaires sont accompagnés d'un complément d'objet, il faut utiliser obligatoirement l'auxiliaire « avoir ». Au contraire, en l'absence de complément d'objet, il faut utiliser l'auxiliaire « être ». Le Tableau 3.19 fournit des exemples d'emploi des auxiliaires pour certains verbes, dans le cas du passé composé. Les banques de données de verbes utilisées pour ce projet fournissent, pour chaque verbe, les auxiliaires à employer.

Dans le cas des temps composés, si un verbe se conjuguant avec l'un ou l'autre des auxiliaires est sélectionné, le hasard détermine lequel des deux auxiliaires est utilisé. Arbitrairement, un poids égal est ici accordé aux deux options.

Une fois le temps de verbe sélectionné, on choisit la personne parmi les six possibilités (1ère, 2e et 3e personnes du singulier ou du pluriel). La personne est déterminée au hasard, avec des poids déterminés selon les fréquences observées dans le corpus de référence. Mais tout comme pour les temps de verbes, l'algorithme offre l'option d'assigner un poids minimal à toutes les personnes, pour s'assurer au besoin qu'elles soient toutes représentées parmi les phrases générées aléatoirement. Il faut cependant tenir compte du fait que l'impératif ne permet que trois personnes (2e du singulier, 1ère du pluriel et 2e du pluriel). De même, certains verbes sont « impersonnels », c'est-à-dire qu'ils ne se conjuguent qu'à la troisième personne du singulier. Par exemple, le verbe « falloir » (« Il *faut* que tu m'aides »). On pense aussi à certains autres verbes comme « neiger » (« Il neige »). Cependant, pour « neiger » comme pour d'autres verbes comme « pleuvoir », on peut employer le verbe à d'autres personnes, mais dans un sens figuré (« des injures pleuvent sur elle »). Dans le cadre du présent projet, par souci de simplicité, un verbe est listé soit comme impersonnel, soit comme non-impersonnel. Les verbes « pleuvoir » et « neiger » ne peuvent donc apparaître dans les phrases générées aléatoirement à l'Étape 2 qu'à la forme impersonnelle, peu importe leur usage observé dans le corpus de référence.

Une fois la personne déterminée, le « nombre » associé au verbe, singulier ou pluriel, s'en trouve automatiquement déterminé aussi. Cette information est critique pour bâtir plus tard le groupe du sujet, quand le verbe est à la troisième personne. La formation du groupe du sujet est discutée plus en profondeur à la Section 3.5.3.2.

Le « genre » du verbe, n'a généralement pas besoin d'être déterminé. Par exemple, à la première personne, il n'importe pas que le sujet « je » soit un homme ou une femme, dans une phrase telle que « je mange ». Et à la troisième personne, que le sujet soit masculin ou féminin n'influence pas la conjugaison. En effet, « il mange » se conjugue de la même façon que « elle mange », de même que « elles mangent » et « ils mangent ». Les mêmes observations s'appliquent pour les autres personnes. Le genre du sujet, s'il ne joue pas un rôle dans la conjugaison, influence toutefois le complément dans le cas d'un verbe attributif. Par exemple, dans « elle semble fâchée » et « il semble fâché », le verbe « sembler » se conjugue de la même façon, mais l'adjectif « fâché » doit s'accorder avec le sujet, en genre et en nombre. La formation du groupe du complément est discutée plus en profondeur à la Section 3.5.3.3.

Le genre joue en revanche un rôle important dans le groupe du verbe pour les temps composés, ceux-ci faisant appel au participe passé. En effet, le participe passé, lorsqu'il s'accorde, doit s'accorder en genre et en nombre avec le sujet. Une fois la personne déterminée au hasard, le nombre l'est aussi, tel que discuté plus haut. Cependant, le genre du participe passé est alors arbitraire, puisque c'est le groupe du verbe qui est formé en premier pour ce projet. Pour les temps composés, les seuls nécessitant un participe passé, le genre doit donc être déterminé au hasard, avec ici des probabilités égales pour le masculin et le féminin. Mais une fois le genre déterminé,

cette information doit être disponible pour la formation des autres composantes de la phrase, pour en assurer l'accord. Ainsi, si le féminin est choisi et qu'on aboutit au groupe du verbe « est arrivée », il faudra s'assurer d'utiliser un nom ou un pronom féminin singulier pour le sujet (« elle est arrivée »).

Dans le cas des participes passés conjugués avec l'auxiliaire « avoir », ceux-ci s'accordent avec le complément d'objet direct si et seulement si celui-ci est situé avant le verbe, par exemple dans la phrase « Les livres que j'ai lus. » Le genre et le nombre dépendent donc ici du genre et du nombre du complément. Cette règle concernant l'utilisation de l'auxiliaire « avoir » sera discutée plus en détail à la Section 3.5.3.3.

Certains verbes s'emploient à la forme pronominale. Sous cette forme, le verbe doit être obligatoirement accompagné d'un pronom personnel réfléchi (accusatif), à la même personne que le sujet du verbe et qui se place devant le verbe (« je me lave »). Toutes les formes pronominales conjuguées à un temps composé utilisent l'auxiliaire « être » (« je me *suis* promené »). Certains verbes sont uniquement pronominaux, tels que « s'écrier », « s'enfuir », et « s'évanouir ». D'autres verbes existent à la fois aux formes non pronominales et pronominales. La forme pronominale nuance ou modifie complètement le sens du verbe, par exemple dans « passer tout droit » et « se passer de quelque chose ». Tout comme pour l'auxiliaire, l'existence de la forme pronominale est fournie pour chacun des verbes de la liste du Bescherelle. Si un verbe peut se retrouver sous l'une ou l'autre des formes, c'est encore le hasard qui détermine laquelle est employée dans la phrase générée aléatoirement.

L'algorithme permet aussi la génération de phrases négatives. Celles-ci sont typiquement caractérisées par l'emploi des adverbes « ne » et « pas » de part et d'autre du verbe conjugué, ou par la présence de ces deux adverbes devant un infinitif. C'est le hasard qui détermine si une phrase générée à l'Étape 2 est sous la forme positive ou négative. Le poids de chacun des cas est choisi arbitrairement et n'est donc pas ici basé sur la fréquence de phrases négatives dans le corpus. Il est aussi à noter que par simplicité, le négatif est toujours formé avec l'adverbe « pas » pour les phrases générées, bien que d'autres possibilités existent en général (« point », « plus », « guère »).

L'algorithme permet aussi l'utilisation de verbes dits « modaux » devant le verbe principal déterminé au hasard. Les verbes modaux utilisés pour ce projet sont listés au Tableau 3.20. Il est à noter qu'aux temps composés, les verbes modaux se conjuguent tous avec l'auxiliaire « avoir », sauf le verbe « aller », qui se conjugue avec « être ». Le verbe « aller » n'est généralement pas considéré comme un verbe modal, mais comme on construit ainsi le futur proche sur le même modèle qu'avec les verbes modaux, le verbe « aller » a été ici considéré comme tel.

Tableau 3.19 : Exemples d'emploi des auxiliaires « avoir » et « être » au passé composé

Auxiliaire(s)	Exemple de phrases avec conjugaison au passé composé
avoir	J'ai mangé un gâteau.
être	Tu es allée à l'épicerie.
avoir ou être	J'ai descendu l'escalier. (complément d'objet) Je suis descendu au sous-sol. (pas un complément d'objet)

Tableau 3.20 : Liste des verbes modaux utilisés dans ce projet

aimer	paraître	sembler
croire	penser	vouloir
devoir	pouvoir	aller
espérer	savoir	

En présence d'un verbe modal, le verbe suivant est employé à l'infinitif, comme dans « je dois *manger* ». L'algorithme peut conjuguer le verbe modal à tous les autres temps, et dans tous les cas, le verbe suivant demeure à l'infinitif. On a par exemple : « j'aurais dû manger », « tu voudras manger », « qu'il veuille manger ». L'utilisation d'un verbe modal est aussi déterminée au hasard, en utilisant une fois de plus un poids arbitraire, non basé sur la fréquence de verbes modaux dans le corpus de référence, par simplicité.

Finalement, le groupe du verbe peut aussi optionnellement contenir des adverbes. Bien qu'il existe plusieurs types d'adverbes (listés au Tableau 3.2), seuls deux types sont considérés ici pour la génération de phrases au hasard : les adverbes de caractérisation et les adverbes d'intensité. Les adverbes de caractérisation incluent entre autres les formes se terminant en « ment », qui dénotent la manière de l'action (« lentement », « fortement », etc.) Les adverbes d'intensité quant à eux, comme leur nom l'indique, précisent l'intensité du mot qu'ils accompagnent. Dans le cas du groupe du verbe *dans ce projet*, trois cas sont possibles :

1. Aucun adverbe utilisé (« je mange »)
2. Un seul adverbe utilisé – adverbe de caractérisation (« je mange lentement »)
3. Deux adverbes utilisés – un adverbe d'intensité suivi d'un adverbe de caractérisation (« je mange très lentement »)

Un poids est assigné arbitrairement à ces trois possibilités, non basé sur les fréquences d'apparition des adverbes dans les groupes du verbe du corpus de référence.

En résumé, la structure et le contenu du groupe du verbe dans les phrases générées au hasard pour l'Étape 2, dépendent des entrées et paramètres listés au Tableau 3.21.

Tableau 3.21 : Entrées et paramètres du groupe du verbe

Entrée ou paramètre	Comment on le détermine	Source
infinitif du verbe	Aléatoire, selon la fréquence d'apparition	Corpus de référence
temps du verbe	Aléatoire, selon la fréquence d'apparition – mais un minimum peut être imposé	Corpus de référence
personne	Aléatoire, selon la fréquence d'apparition – mais un minimum peut être imposé	Corpus de référence
auxiliaire pour les temps composés	Associé à chaque infinitif. Trois options possibles : • être • avoir • soit l'un soit l'autre	Liste des verbes du Bescherelle
forme pronominale	Associée à chaque infinitif. Trois options possibles : • non permise • optionnelle • exclusive Si optionnelle, déterminée aléatoirement selon un poids arbitraire	Liste des verbes du Bescherelle
verbe impersonnel	Associé à chaque infinitif. Trois options généralement possibles : • verbe non impersonnel • verbe exclusivement impersonnel • verbe impersonnel ou non Pour ce projet, si un verbe est classé « impersonnel », il ne sera pas employé autrement	Liste des verbes du Bescherelle
forme modale	Sélectionnée aléatoirement selon un poids arbitraire	Liste exhaustive des verbes modaux de la langue française, en plus du verbe « aller »
présence d'adverbes	Trois options considérées : • aucun adverbe • adverbe de caractérisation seul • adverbe d'intensité suivi d'un adverbe de caractérisation Déterminée aléatoirement selon un poids arbitraire	Adverbes de caractérisation selon le corpus de référence et cooccurrences Adverbes d'intensité selon une base de données plus complète indépendante du corpus

3.5.3.1.1. Séquences des mots du groupe du verbe

Une fois toutes les entrées et tous les paramètres du Tableau 3.21 déterminés par l'algorithme, le groupe du verbe peut être construit. Dépendamment des mots en jeu (formes conjuguées, verbes modaux, forme négative ou positive, pronoms réfléchis, adverbes, etc.) plusieurs séquences de mots sont possibles. Celles-ci sont regroupées en huit grandes catégories, toutes décrites plus bas. En tout, 108 possibilités de séquences de mots pour le groupe du verbe peuvent être générées par l'algorithme développé pour ce projet.

Forme 1 : *Impératif négatif*

Notes :

- Les parenthèses carrées indiquent que la présence du mot est optionnelle.
- Pour la forme négative, on utilise toujours pour ce projet « ne » et « pas ».
- Le pronom réfléchi n'est utilisé que pour les formes pronominales. Les pronoms à utiliser sont : « te », « nous », « vous » (seules trois personnes possibles à l'impératif).
- Comme le verbe peut être pronominal ou non (2 options) et qu'il y a trois options pour les adverbes (pas d'adverbe, un seul adverbe ou deux adverbes), il y a en tout 2 x 3 = 6 possibilités de constructions de phrases pour la Forme 1, toutes listées plus bas.

Exemples :

- « Ne mange pas », « Ne mange pas vite », « Ne mange pas trop vite ».
- « Ne nous lavons pas », « Ne nous lavons pas vite », « Ne nous lavons pas trop vite ».

Forme 2 : *Impératif positif*

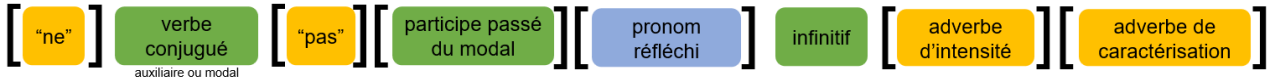
Notes :

- Le pronom réfléchi n'est utilisé que pour les formes pronominales.
  - En contraste avec la forme négative, le pronom réfléchi est placé après le verbe conjugué.
  - Les pronoms à utiliser sont : « toi », « nous », « vous » (seules trois personnes sont possibles à l'impératif). Le pronom pour la deuxième personne du singulier est donc différent de celui pour la forme négative.
  - Un trait d'union est nécessaire pour relier le verbe conjugué et le pronom réfléchi.
- Comme le verbe peut être pronominal ou non (2 options) et qu'il y a trois options pour les adverbes (pas d'adverbe, un seul adverbe ou deux adverbes), il y a en tout 2 x 3 = 6 possibilités de constructions de phrases pour la Forme 2, toutes listées plus bas.

Exemples :

- « Mange », « Mange vite », « Mange très vite »
- « Lavez-vous », « Lavez-vous vite », « Lavez-vous très vite »

Forme 3 : *Avec verbe modal – adverbe(s) à la fin du groupe verbe*



Notes :

- Dans le cas de temps composés, le verbe conjugué est l'auxiliaire employé avec le verbe modal (toujours l'auxiliaire « avoir », sauf pour le verbe « aller », considéré pour ce projet comme un verbe modal).

- Pour les temps non-composés, le verbe conjugué est le verbe modal.
- Pour les temps composés, le participe passé du modal est utilisé.
- Considérant que le groupe du verbe peut être positif ou négatif (2 options), que le temps peut être composé ou non (2 options), que le verbe peut être pronominal ou non (2 options) et qu'il y a trois options pour les adverbes (pas d'adverbe, un seul adverbe ou deux adverbes), il y a en tout 2 x 2 x 2 x 3 = 24 possibilités de constructions de phrases, toutes listées plus bas.
- Dans les exemples plus bas, un pronom personnel sujet est fourni en italique, mais celui-ci fait partie du groupe du sujet, et non pas du groupe du verbe. Il n'est ajouté ici que pour faciliter la lecture.

Exemples :

- « *Il* veut manger », « *Il* veut manger vite », « *Il* veut manger très vite »
- « *Il* ne veut pas manger », « *Il* ne veut pas manger vite », « *Il* ne veut pas manger très vite »
- « *Nous* avons voulu manger », « *Nous* avons voulu manger vite », « *Nous* avons voulu manger très vite »
- « *Nous* n'avons pas voulu manger », « *Nous* n'avons pas voulu manger vite », « *Nous* n'avons pas voulu manger très vite »
- « *Ils* doivent se rendre », « *Ils* doivent se rendre vite », « *Ils* doivent se rendre très vite »
- « *Ils* ne doivent pas se rendre », « *Ils* ne doivent pas se rendre vite », « *Ils* ne doivent pas se rendre très vite »
- « *J'*ai dû me rendre », « *J'*ai dû me rendre vite », « *J'*ai dû me rendre très vite »
- « *Je* n'ai pas dû me rendre », « *Je* n'ai pas dû me rendre vite », « *Je* n'ai pas dû me rendre très vite »

Forme 4 : *Avec verbe modal – adverbe(s) avant le pronom réfléchi (le cas échéant) et l'infinitif*

Notes :

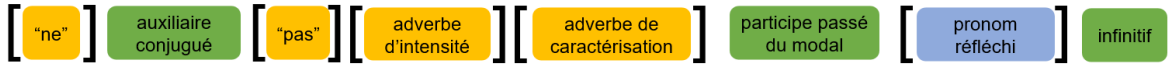
- Cette forme est identique à la Forme 3, à l'exception de la position des adverbes, lorsque présents. On retrouve donc les mêmes 24 possibilités, toutes listées plus bas.
- Les mêmes exemples qu'à la Forme 3 ont été utilisés, pour clairement indiquer la différence entre ces deux formes. Les exemples sans adverbe (le premier exemple sur chaque ligne) sont identiques pour les Formes 3 et 4.
- Dans les exemples plus bas, un pronom personnel sujet est fourni en italique, mais celui-ci fait partie du groupe du sujet, et non pas du groupe du verbe. Il n'est ajouté ici que pour faciliter la lecture.

Exemples :

- « *Il* veut manger », « *Il* veut vite manger », « *Il* veut très vite manger »
- « *Il* ne veut pas manger », « *Il* ne veut pas vite manger », « *Il* ne veut pas très vite manger »
- « *Nous* avons voulu manger », « *Nous* avons voulu vite manger vite », « *Nous* avons voulu très vite manger »
- « *Nous* n'avons pas voulu manger », « *Nous* n'avons pas voulu vite manger », « *Nous* n'avons pas voulu très vite manger »
- « *Ils* doivent se rendre », « *Ils* doivent vite se rendre », « *Ils* doivent très vite se rendre »

- « *Ils* ne doivent pas se rendre », « *Ils* ne doivent pas vite se rendre vite », « *Ils* ne doivent pas très vite se rendre »
- « *J'*ai dû me rendre », « *J'*ai dû vite me rendre », « *J'*ai très vite me rendre »
- « *Je* n'ai pas dû me rendre », « *Je* n'ai pas dû vite me rendre », « *Je* n'ai pas dû très vite me rendre »

Forme 5 : *Avec verbe modal – adverbe(s) avant le participe passé du modal (temps composés)*



Notes :

- Cette forme est très semblable aux Formes 3 et 4, à l'exception de la position des adverbes, lorsque présents. Aussi, on se limite ici uniquement aux temps composés.
- Considérant que le groupe verbe peut être positif ou négatif (2 options), que le verbe peut être pronominal ou non (2 options) et qu'il y a trois options pour les adverbes (pas d'adverbe, un seul adverbe ou deux adverbes), il y a en tout 2 x 2 x 3 = 12 possibilités de constructions de phrases, toutes listées plus bas.
- Les mêmes exemples qu'aux Formes 3 et 4 sont utilisés, mais uniquement les formes aux temps composés, puisque ce sont les seules applicables ici.
- Les exemples sans adverbe (le premier exemple sur chaque ligne) sont identiques pour les Formes 3, 4 et 5.
- Dans les exemples plus bas, un pronom personnel sujet est fourni en italique, mais celui-ci fait partie du groupe du sujet, et non pas du groupe du verbe. Il n'est ajouté ici que pour faciliter la lecture.

Exemples :

- « *Nous* avons voulu manger », « *Nous* avons vite voulu manger », « *Nous* avons très vite voulu manger »
- « *Nous* n'avons pas voulu manger », « *Nous* n'avons pas vite voulu manger », « *Nous* n'avons pas très vite voulu manger »
- « *J'*ai dû me rendre », « *J'*ai vite dû me rendre vite », « *J'*ai très vite dû me rendre »
- « *Je* n'ai pas dû me rendre », « *Je* n'ai pas vite dû me rendre », « *Je* n'ai pas très vite dû me rendre »

Forme 6 : *Ni impératif, ni modal – Temps composé et adverbe après le participe passé*

Notes :

- Considérant que le groupe verbe peut être positif ou négatif (2 options), que le verbe peut être pronominal ou non (2 options) et qu'il y a trois options pour les adverbes (pas d'adverbe, un seul adverbe ou deux adverbes), il y a en tout 2 x 2 x 3 = 12 possibilités de constructions de phrases, toutes listées plus bas.
- Dans les exemples plus bas, un pronom personnel sujet est fourni en italique, mais celui-ci fait partie du groupe du sujet, et non pas du groupe du verbe. Il n'est ajouté ici que pour faciliter la lecture.

Exemples :

- « *Il* a mangé. », « *Il* a mangé vite. », « *Il* a mangé très vite. »
- « *Il* n'a pas mangé. », « *Il* n'a pas mangé vite. », « *Il* n'a pas mangé très vite. »
- « *Il* s'est lavé », « *Il* s'est lavé vite », « *Il* s'est lavé très vite »
- « *Il* ne s'est pas lavé », « *Il* ne s'est pas lavé vite », « *Il* ne s'est pas lavé très vite »

Forme 7 : *Ni impératif, ni modal – Temps composé et adverbe devant le participe passé*

Notes :

- Cette forme est identique à la Forme 6, à l'exception de la position des adverbes, lorsque présents. On retrouve donc les mêmes 12 possibilités, toutes listées plus bas.
- Les mêmes exemples qu'à la Forme 6 ont été utilisés, pour clairement indiquer la différence entre ces deux formes. Les exemples sans adverbe (le premier exemple sur chaque ligne) sont identiques pour les Formes 6 et 7.
- Dans les exemples plus bas, un pronom personnel sujet est fourni en italique, mais celui-ci fait partie du groupe du sujet, et non pas du groupe du verbe. Il n'est ajouté ici que pour faciliter la lecture.

Exemples :

- « *Il* a mangé. », « *Il* a vite mangé. », « *Il* a très vite mangé. »
- « *Il* n'a pas mangé. », « *Il* n'a pas vite mangé. », « *Il* n'a pas très vite mangé. »
- « *Il* s'est lavé », « *Il* s'est vite lavé », « *Il* s'est très vite lavé »
- « *Il* ne s'est pas lavé », « *Il* ne s'est pas vite lavé », « *Il* ne s'est pas très vite lavé »

Forme 8 : *Ni impératif, ni modal – Temps non composé*

Notes :

- Considérant que le groupe verbe peut être positif ou négatif (2 options), que le verbe peut être pronominal ou non (2 options) et qu'il y a trois options pour les adverbes (pas d'adverbe, un seul adverbe ou deux adverbes), il y a en tout 2 x 2 x 3 = 12 possibilités de constructions de phrases, toutes listées plus bas.
- Dans les exemples plus bas, un pronom personnel sujet est fourni en italique, mais celui-ci fait partie du groupe du sujet, et non pas du groupe du verbe. Il n'est ajouté ici que pour faciliter la lecture.

Exemples :

- « *Il* mange », « *Il* mange vite », « *Il* mange très vite »
- « *Il* ne mange pas », « *Il* ne mange pas vite », « *Il* ne mange pas très vite »
- « *Elle* se promène », « *Elle* se promène vite », « *Elle* se promène très vite »
- « *Elle* ne se promène pas », « *Elle* ne se promène pas vite », « *Elle* ne se promène pas très vite »

Tableau 3.22 : Éléments en sortie de l'algorithme du groupe du verbe avec exemples pour le groupe du verbe « n'aurait pas voulu manger vite »

Éléments en sortie	Exemple
Liste des mots (fléchis)	« n' », « aurait », « pas », « voulu », « manger », « vite »
Liste des lemmes	« ne », « avoir », « pas », « vouloir », « manger », « vite »
Liste des classes grammaticales	adverbe, verbe, adverbe, verbe, verbe, adverbe
Paramètres morpho-syntaxiques	ne : adverbe aurait : verbe avoir, conditionnel présent, 3e personne du singulier pas : adverbe voulu : verbe vouloir, participe passé, masculin singulier manger : verbe manger, infinitif vite : adverbe de caractérisation
Tableaux de cooccurrences	« ne » « avoir » « pas » « vouloir » « manger » « vite »
Personne du verbe	Troisième du singulier
Temps du verbe	Conditionnel passé
Intransitif / transitif	Transitif ou intransitif
Verbe attributif?	Non
Verbe impersonnel?	Non
Genre	Non applicable
Nombre	Singulier
Utilisation de préposition	Aucune

3.5.3.1.2. Informations fournies en sortie par l'algorithme du groupe du verbe

Une fois tous les mots du groupe du verbe déterminés, selon les paramètres du Tableau 3.21 et les structures discutées à la Section 3.5.3.1.1, certaines informations doivent être fournies en sortie, afin d'accorder le groupe du sujet avec le groupe du verbe, et afin de bien arrimer le groupe du complément au groupe du verbe. Ultimement, la phrase complète sera aussi formée. Le Tableau 3.22 fournit une liste de tous les éléments retournés en sortie par l'algorithme du groupe du verbe. Des exemples sont fournis pour un groupe du verbe particulier déterminé arbitrairement.

3.5.3.2. Génération du groupe du sujet

Il faut s'assurer que le groupe du sujet s'accorde avec le groupe du verbe. Pour ce faire, on doit faire appel à certains paramètres fournis en sortie par l'algorithme du groupe du verbe listés au Tableau 3.22. Tout d'abord, la personne du groupe du sujet doit s'accorder avec la personne du groupe du verbe. Si par exemple le groupe du verbe est à la deuxième personne du singulier (« manges »), le groupe du sujet doit lui aussi être à la deuxième personne du singulier (« *tu* manges »). Si le groupe du verbe implique un verbe impersonnel, le groupe du sujet doit obligatoirement faire appel au pronom personnel « il » comme sujet du verbe. On ne pourrait par exemple utiliser ici un nom commun. On peut écrire « il pleut », mais non « la table pleut ».

Le groupe sujet doit aussi s'accorder en nombre avec le groupe du verbe, le nombre étant déterminé par la personne du verbe. En effet, les trois personnes du singulier requièrent un sujet au singulier tandis que les trois personnes du pluriel requièrent un sujet au pluriel. Il en est de même pour le genre. Cependant, dans la plupart des cas, le genre n'est pas imposé par le groupe du verbe. En effet, si par exemple le groupe du verbe est « mange de la crème glacée », bien que le nombre soit déterminé (singulier), le genre ne l'est pas. On peut écrire « il mange de la crème glacée » aussi bien que « elle mange de la crème glacée ». C'est uniquement en présence d'un participe passé conjugué avec l'auxiliaire être que le genre est imposé au sein du groupe du verbe.

Par exemple, si le groupe du verbe est « sont arrivées », il va de soi que le sujet devra être lui aussi au féminin pluriel. Dans tous les autres cas, n'impliquant donc pas le participe passé, le genre du sujet doit être déterminé aléatoirement au sein du groupe du sujet.

De façon générale, le temps du verbe n'influe pas sur le groupe du sujet. En effet, on peut par exemple utiliser le même pronom personnel sujet « il » que le verbe soit conjugué à n'importe lequel de ces temps :

- Il veut
- Il voulait
- Il voudrait
- Il aurait voulu

Il y a en revanche deux temps de verbe nécessitant un traitement particulier. Tout d'abord, l'impératif est le plus simple à considérer, car celui-ci s'utilise sans groupe du sujet. Ainsi, si le groupe du verbe est « ne mange pas trop vite » où « mange » est ici une forme impérative, on ne peut pas ajouter de groupe du sujet. Dans un tel cas, la phrase, en supposant pour l'instant l'absence de groupe du complément, demeure tout simplement « Ne mange pas trop vite ». Le groupe du sujet ici demeure donc vide.

L'autre temps de verbe nécessitant un traitement particulier est le subjonctif. Il existe plusieurs tournures de phrases pouvant accompagner un verbe conjugué à un temps du subjonctif. Mais pour le travail actuel, à chaque fois qu'un subjonctif est généré au sein du groupe du verbe, on impose la forme « il faut que » pour l'introduire. Par exemple, si le groupe du verbe est « viennes très rapidement », le groupe du sujet sera « il faut que tu », donnant la phrase « Il faut que tu viennes très rapidement ». Dans une analyse syntaxique traditionnelle, la locution « il faut que » n'est pas partie prenante du groupe du sujet. Mais par convenance et simplicité, pour le projet actuel, elle en fera partie. Le temps de verbe utilisé pour le verbe falloir dans la locution « il faut que » dépend du temps utilisé dans le groupe du verbe. Si le verbe du groupe du verbe est au subjonctif présent ou au subjonctif imparfait, seuls certains temps du verbe « falloir » sont alors permis, tels que listés au Tableau 3.23. Si en revanche le verbe du groupe du verbe est au subjonctif passé ou subjonctif plus-que-parfait, seul l'imparfait peut être utilisé pour le verbe « falloir ». On a donc « il fallait qu'il ait mangé » et « il fallait qu'il eût mangé ».

Tableau 3.23 : Temps de verbe permis pour la locution « il faut que », si le groupe verbe est au subjonctif présent ou au subjonctif imparfait, avec exemples

Temps du verbe « falloir »	Exemples
Présent	Il faut qu'il mange. Il faut qu'il mangeât.
Imparfait	Il fallait qu'il mange. Il fallait qu'il mangeât.
Passé simple	Il fallut qu'il mange. Il fallut qu'il mangeât.
Futur simple	Il faudra qu'il mange. Il faudra qu'il mangeât.
Conditionnel	Il faudrait qu'il mange. Il faudrait qu'il mangeât.
Passé composé	Il a fallu qu'il mange. Il a fallu qu'il mangeât.
Plus-que-parfait	Il avait fallu qu'il mange. Il avait fallu qu'il mangeât.
Passé antérieur	Il eut fallu qu'il mange. Il eut fallu qu'il mangeât.
Futur antérieur	Il aura fallu qu'il mange. Il aura fallu qu'il mangeât.
Conditionnel passé	Il aurait fallu qu'il mange. Il aurait fallu qu'il mangeât.

Tableau 3.24 : Éléments en sortie de l'algorithme du groupe du verbe requis pour la génération du groupe du sujet

Éléments en sortie du groupe du verbe	Utilité pour le groupe du sujet
Personne du verbe	La personne du groupe du sujet doit s'accorder avec la personne du groupe du verbe. Pour ce projet, seules les 3e personnes du singulier et du pluriel permettent l'emploi de noms communs pour le groupe sujet.
Verbe impersonnel (vrai ou faux)	Si le verbe est impersonnel, le sujet devient impérativement « il ».
Nombre	Le nombre du groupe du verbe dépend de la personne du verbe (les trois premières personnes demandent le singulier, les trois suivantes le pluriel). Le groupe du sujet doit donc ici s'accorder en nombre avec le groupe du verbe, celui-ci étant déterminé en premier.
Genre	Dans les cas impliquant le participe passé, le genre est ici déterminé au sein du groupe du verbe. Le même genre doit être utilisé pour le groupe du sujet. Mais dans les autres cas, le genre est déterminé aléatoirement au sein du groupe du sujet.
Temps du verbe	Si le verbe est à l'impératif, le groupe du sujet reste vide. Si le verbe est au subjonctif, on inclut une tournure de phrase du type « il faut que ». Pour tous les autres cas, le temps du verbe ne joue aucun rôle pour déterminer le groupe du sujet.
Mots pour tableaux de cooccurrences	On détermine au hasard les lemmes à employer dans le groupe du sujet parmi les cooccurrences des mots utilisés jusqu'à présent dans le groupe du verbe.

Finalement, la dernière information du groupe du verbe nécessaire pour la formation du groupe du sujet est la liste des noms communs, adjectifs, verbes et adverbes utilisés dans le groupe du verbe. On extrait des mots issus de ces quatre classes grammaticales toutes leurs cooccurrences retrouvées dans le corpus de référence. C'est en priorité parmi ces cooccurrences que les noms et adjectifs du groupe du sujet sont choisis. Par exemple, si le verbe « jouer » est utilisé dans le groupe du verbe et qu'on retrouve le nom commun « enfant » parmi les cooccurrences de ce verbe dans le corpus de référence, le nom « enfant » devient alors un des candidats pour la sélection d'un nom commun dans le groupe du sujet. Le sous-ensemble de paramètres du groupe du verbe dont il a été question ici, nécessaire pour la création du groupe du sujet, est résumé au Tableau 3.24.

Une fois que tous les éléments requis du groupe du verbe sont déterminés, on peut procéder à la construction du groupe du sujet. Il faut d'abord noter que deux types de sujets sont possibles : ceux impliquant des noms communs et ceux impliquant des pronoms. Par simplicité, les noms propres ont été exclus. Pour chaque phrase à générer au hasard, il faut donc déterminer si le groupe du sujet sera bâti à partir de noms communs ou de pronoms. Cependant, la personne du verbe impose certaines contraintes quant à ces deux possibilités, tel qu'on le résume au Tableau 3.25.

Tableau 3.25 : Utilisation de noms communs ou de pronoms pour le groupe du sujet, en fonction de la personne du verbe

Personne du verbe	Exemples généraux	Exemples pour ce travail
1ère du singulier	Je mange.	Je mange.
2e du singulier	Tu manges.	Tu manges.
3e du singulier	Il mange. La fille mange.	Il mange. La fille mange.
1ère du pluriel	Nous mangeons. L'homme et moi mangeons.	Nous mangeons.
2e du pluriel	Vous mangez. Les amis et toi mangez.	Vous mangez.
3e du pluriel	Ils mangent. Les enfants mangent.	Ils mangent. Les enfants mangent.

Comme on le constate à la deuxième colonne du Tableau 3.25, il est possible de façon générale de générer aux deux premières personnes du pluriel des groupes du sujet impliquant des noms communs. Cependant, par simplicité et convenance, pour ce mémoire, seules les troisièmes personnes du pluriel et du singulier impliqueront l'utilisation de noms communs dans les groupes du sujet. Autrement dit, pour ce projet, lorsqu'il est question des deux premières personnes du singulier et du pluriel, l'algorithme ne permet que l'utilisation de pronoms.

Aux sections suivantes, on présente donc comment les groupes du sujet sont générés aléatoirement, dans le cas où on utilise des pronoms (Section 3.5.3.2.1) et des noms communs (Section 3.5.3.2.2).

3.5.3.2.1. Génération du groupe sujet – avec pronoms

Tous les groupes du verbe, à l'exception du cas impératif, peuvent accepter des pronoms pour le groupe du sujet. Les pronoms les plus simples à utiliser sont les pronoms personnels sujet « je », « tu », « il/elle », « on », « nous », « vous », et « ils/elles ». Dans ces cas, la sélection du pronom se fait de façon très simple : il suffit d'accorder la personne du pronom avec celle du groupe du verbe, tel que montré au Tableau 3.26. À la troisième personne, au pluriel comme au singulier, on constate donc que plus d'une option est offerte. C'est le hasard qui déterminera laquelle de ces options sera sélectionnée.

Tableau 3.26 : Utilisation des pronoms personnels sujet en fonction de la personne du groupe du verbe

Personnes du groupe du verbe	Pronoms personnels sujet possibles
1ère du singulier	je
2e du singulier	tu
3e du singulier	il, elle, on
1ère du pluriel	nous
2e du pluriel	vous
3e du pluriel	ils, elles

Tableau 3.27 : Liste des pronoms possessifs (accompagnés d'un déterminant)

Genre et nombre	1ère personne du singulier	2e personne du singulier	3e personne du singulier	1ère personne du pluriel	2e personne du pluriel	3e personne du pluriel
Masculin singulier	le mien	le tien	le sien	le nôtre	le vôtre	le leur
Masculin pluriel	les miens	les tiens	les siens	les nôtres	les vôtres	les leurs
Féminin singulier	la mienne	la tienne	la sienne	la nôtre	la vôtre	la leur
Féminin pluriel	les miennes	les tiennes	les siennes	les nôtres	les vôtres	les leurs

Aux premières et deuxièmes personnes, singulier comme pluriel, les seules options de pronoms possibles pour ce projet sont celles du Tableau 3.26. Cependant, pour la troisième personne du singulier ou la troisième personne du pluriel, d'autres types de pronoms peuvent être utilisés au sein du groupe du sujet. On peut par exemple employer des pronoms possessifs. Le Tableau 3.27 en fournit la liste exhaustive pour ce projet.

Il faut se rappeler que les pronoms possessifs ne sont employés que lorsque le groupe du verbe est à la troisième personne du singulier ou du pluriel. Il est donc important de noter que les personnes listées à la première ligne du Tableau 3.27 ne correspondent pas à la personne du groupe du verbe, mais plutôt à celle à qui appartient ce dont on parle. Du groupe du verbe, on se contente donc d'extraire le nombre, qui dépend de la personne du verbe, et le genre. Tel que mentionné lors de la discussion du groupe du verbe, le genre est dans la plupart des cas arbitraire et peut donc être déterminé au hasard pour le groupe du sujet. La personne, telle que listée à la première ligne du Tableau 3.27 est donc tout à fait arbitraire dans le contexte des phrases générées aléatoirement. C'est donc au hasard qu'une telle personne sera déterminée pour sélectionner un pronom possessif.

D'autres types de pronoms peuvent être utilisés pour le groupe du sujet. L'algorithme actuel inclut en effet ces autres possibilités : « tous/toutes », « celui-ci/celle-ci », « celui-là/celle-là », « personne », « quelqu'un / quelques-uns / quelques-unes », « rien », « quiconque », « n'importe lequel / laquelle / lesquels / lesquelles ». Pour ces formes, tout comme pour les pronoms personnels sujet du Tableau 3.26 et les pronoms possessifs du Tableau 3.27, il faut s'assurer d'accorder le genre et le nombre avec le groupe du verbe. Encore là, le genre est souvent arbitraire. Mais pour ces formes de pronoms additionnelles, il existe certaines contraintes, en fonction du nombre associé au groupe du verbe, ou selon que le groupe verbe est à la forme affirmative ou négative. Ces contraintes sont listées au Tableau 3.28.

Tableau 3.28 : Liste des pronoms et leurs contraintes d'utilisation selon les paramètres du groupe du verbe qu'ils accompagnent

Type de pronom	Pluriel	Singulier	Forme affirmative	Forme négative
personnel-sujet (je, tu, il…)	✓	✓	✓	✓
possessif (le mien, le tien…)	✓	✓	✓	✓
tous / toutes	✓	✗	✓	✗
celui-ci, celui-là, celle-ci, celle-là	✓	✓	✓	✓
personne	✗	✓	✗	✓
quelqu'un, quelques-uns, quelques-unes	✓	✓	✓	✗
rien	✗	✓	✗	✓
quiconque	✗	✓	✓	✓
n'importe lequel / laquelle / lesquels / lesquelles / qui	✓	✓	✓	✗

Les contraintes reliées au nombre listées au Tableau 3.28 s'expliquent aisément. En effet, « tous » et « toutes » sont par définition au pluriel, « personne » est au singulier, de même que « rien » et « quiconque ». Les contraintes reliées aux formes affirmatives et négatives du groupe du verbe sont un peu plus subtiles. Des phases du type « tous ne sont pas » sont possibles, mais paraissent un peu étranges, et sont donc évitées ici pour la formation aléatoire de phrases. Pour ce qui est du mot « personne » employé comme pronom, c'est toujours dans des phrases négatives qu'on le retrouve (« personne ne vient »). De la même façon, on n'emploie le pronom « rien » que pour les formes négatives (« rien ne fonctionne »). Il paraît aussi étrange d'écrire « quelqu'un ne vient pas », ou « n'importe lequel ne convient pas ». Les contraintes listées au Tableau 3.28 ont donc été mises en place pour éviter de générer des phrases au sens douteux ou sonnant « faux ».

Pour générer des groupes du sujet basés sur des pronoms, il faut donc déterminer en premier lieu le type de pronom à employer. Pour les premières et deuxièmes personnes (singulier ou pluriel), le seul choix permis pour ce projet est le pronom personnel sujet (« je », « tu », « nous », « vous »). Pour la troisième personne en revanche, le Tableau 3.28 offre plusieurs options. L'algorithme devra donc pouvoir déterminer au hasard laquelle de ces options sélectionner. Pour ce faire, des poids seront affectés arbitrairement à chacune de ces options, sans égard à leur fréquence d'apparition dans le corpus de référence. On tiendra évidemment compte des contraintes listées au Tableau 3.28 pour éviter l'utilisation de certains pronoms dans certaines circonstances. De plus, il faut s'assurer d'accorder les pronoms en genre et en nombre avec le groupe du verbe. Finalement, pour les pronoms possessifs, la personne à qui appartient ce dont on parle sera aussi déterminée arbitrairement. Cette fois, l'algorithme donne un poids égal, donc une probabilité égale, à chacune des options possibles, dépendamment du nombre, qui lui, est déterminé par le groupe du verbe.

3.5.3.2.2. Génération du groupe sujet – avec noms communs

Tel que mentionné précédemment, les noms communs ne sont utilisés dans le groupe du sujet que lorsque le groupe du verbe est à la troisième personne du singulier ou à la troisième personne du pluriel. Il faut comme toujours s'assurer d'accorder le groupe du sujet avec le groupe du verbe. Le groupe du verbe fournit toujours un nombre à respecter, qui dépend de la personne du verbe. Pour ce qui est du genre, celui-ci est généralement arbitraire, sauf si un participe passé est conjugué avec l'auxiliaire être dans le groupe du verbe. Dans un tel cas, le genre est spécifié au sein du groupe verbe. Si le genre n'est pas spécifié, il est déterminé aléatoirement, avec probabilités égales pour chacun.

Pour le cas singulier, un nom commun au genre approprié est déterminé aléatoirement parmi les cooccurrences des mots faisant partie du groupe du verbe, en fonction de leur fréquence d'apparition parmi ces cooccurrences. Pour le pluriel, on peut choisir un nom commun qu'on utilise au pluriel. Mais l'algorithme offre une autre option : combiner deux noms communs. Dans un cas général, le nombre de noms communs pourrait être infini, mais l'algorithme, par convenance et simplicité, limite ce nombre à seulement deux pour ce projet. L'utilisation de deux noms communs, séparés par la conjonction « et » nous assure que le groupe sujet sera au pluriel. Il faut toutefois s'assurer que le genre sera respecté. Le Tableau 3.29 résume les possibilités ainsi que les contraintes à respecter lors de la sélection aléatoire de deux noms communs dans le groupe du sujet, en fonction de leur genre et de leur nombre.

Lors de la sélection des noms communs parmi les cooccurrences, il faut donc s'assurer de choisir des noms qui répondent aux contraintes du Tableau 3.29 pour le genre et le nombre. Une fois le ou les noms communs choisis, il faut choisir le ou les déterminants qui vont accompagner ces noms, car seuls les noms propres, non utilisés pour ce projet, peuvent être employés sans déterminant dans le groupe du sujet. Toutes les possibilités de déterminants utilisés pour ce projet sont listées au Tableau 3.30, avec certaines contraintes pour leur utilisation, de la même façon qu'on l'a fait pour les pronoms au Tableau 3.28.

Tableau 3.29 : Possibilités et contraintes pour le genre et le nombre des noms communs employés dans le groupe du sujet

Nombre du groupe du verbe	Genre du groupe du verbe	Genre et nombre quand un seul nom commun	Genre et nombre quand deux noms communs	
singulier	masculin	masculin singulier	⊗	⊗
singulier	féminin	féminin singulier	⊗	⊗
pluriel	masculin	masculin pluriel	masculin singulier	masculin singulier
			masculin singulier	masculin pluriel
			masculin pluriel	masculin singulier
			masculin pluriel	masculin pluriel
			masculin singulier	féminin singulier
			masculin singulier	féminin pluriel
			masculin pluriel	féminin singulier
			masculin pluriel	féminin pluriel
			féminin singulier	masculin singulier
			féminin pluriel	masculin singulier
			féminin singulier	masculin pluriel
			féminin pluriel	masculin pluriel
pluriel	féminin	féminin pluriel	féminin singulier	féminin singulier
			féminin singulier	féminin pluriel
			féminin pluriel	féminin singulier
			féminin pluriel	féminin pluriel

Tableau 3.30 : Liste des déterminants et leurs contraintes d'utilisation selon les paramètres du groupe du verbe qu'ils accompagnent

Type de déterminant	Pluriel	Singulier	Forme affirmative	Forme négative	Liste
Article défini	✓	✓	✓	✓	le, la, les
Article indéfini	✓	✓	✓	✓	un, une, des
Démonstratif	✓	✓	✓	✓	ce, cette, ces
Possessif	✓	✓	✓	✓	mon, ton, son, ma, ta, sa, notre, votre, leur, mes, tes, ses, nos, vos, leurs
« chaque »	✗	✓	✓	✗	
« divers »	✓	✗	✓	✓	
« quelques »	✓	✗	✓	✓	
« nul »	✗	✓	✗	✓	
« plusieurs »	✓	✗	✓	✓	
« n'importe quel / quelle »	✗	✓	✓	✗	
« un tel / une telle / de tels / de telles»	✓	✓	✓	✓	
« un/une quelconque »	✗	✓	✓	✗	
« un/une certain/certaine »	✗	✓	✓	✗	
« certains / certaines »	✓	✗	✓	✓	

Il est à noter que pour ce projet, quand deux noms communs sont utilisés pour former le pluriel, le même type de déterminant est utilisé pour les deux noms. On aura par exemple « un tel camion et une telle voiture », mais on ne retrouvera pas « un tel camion et certaines voitures », bien que tout à fait convenable grammaticalement. Ce choix a été fait par simplicité et convenance.

Une fois les noms communs et déterminants choisis, on peut ensuite inclure optionnellement des adjectifs et adverbes. La présence de ces deux classes de mots est basée sur le hasard. On assigne en effet arbitrairement la probabilité d'utilisation d'un adjectif pour modifier le nom, et ensuite la probabilité d'inclure un adverbe d'intensité. De façon générale, on pourrait inclure plus d'un adjectif (« un camion rouge et bleu »), mais par simplicité, un seul adjectif est considéré pour

ce projet pour la génération de phrases aléatoires. L'adjectif doit s'accorder en genre et en nombre avec le nom qu'il modifie. En revanche, si deux noms sont utilisés, chacun peut avoir son propre adjectif (« un camion bleu et une voiture rouge »). Ensuite, chaque adjectif peut être modifié par un adverbe d'intensité (« un camion très lourd »).

3.5.3.2.3. Séquences des mots du groupe du sujet de base

Considérons en premier lieu le groupe sujet dit « de base ». Les variations plus complexes du groupe du sujet incluant des compléments du nom sont introduites plus bas, aux Sections 3.5.3.2.4 à 3.2.5.3.2.6. La séquence des mots du groupe du sujet de base est plus simple que pour le cas du groupe du verbe.

Forme 1 : *Avec pronoms*

Notes :

- On utilise un déterminant uniquement dans le cas des pronoms possessifs (« le mien », « les vôtres »).
- Le pronom ici consiste parfois en une locution (« n'importe lequel ») plutôt qu'un mot unique – se référer au Tableau 3.28.
- Aucun autre mot (adjectif, adverbe) n'est utilisé pour le groupe du sujet dans le cas d'utilisation du pronom.

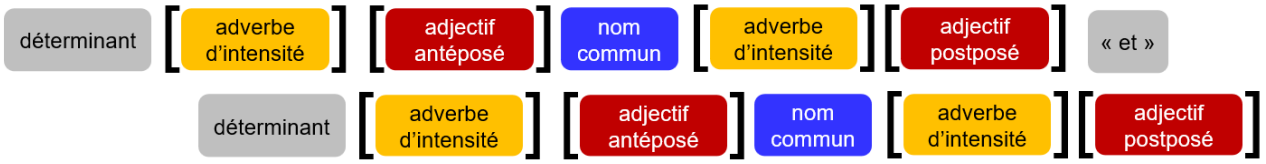
Exemples (incluant le groupe du verbe pour faciliter la lecture en contexte) :

- « *Je* mange beaucoup »
- « *La mienne* fonctionne »
- « *Tous* regardaient attentivement »
- « *Celle-là* conviendrait »
- « *Personne* ne va venir rapidement »
- « *Quelqu'un* aurait voulu »
- « *Rien* ne va » (avec « rien », on n'inclut pas le « pas » pour le négatif)
- « *Quiconque* réussirait »
- « *N'importe lesquels* tomberont »

Forme 2 : *Avec noms communs*

Un seul nom commun (singulier ou pluriel) :

Deux noms communs (pluriel seulement)



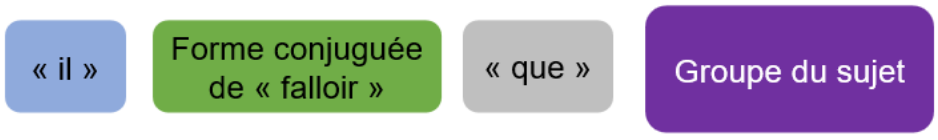
Notes :

- Certains adjectifs sont antéposés – ils peuvent se placer devant le nom (« le gros camion »), d'autres sont postposés (« le camion rouge ») – ils se placent après le nom.
- Pour ce projet, pas plus d'un adjectif accompagne chaque nom, par simplicité. Ainsi, « le gros camion rouge » n'est pas une forme possible pour ce projet, bien que tout à fait convenable grammaticalement.

Exemples (incluant le groupe du verbe) :

- *Le camion* roule.
- *Un gros camion* roule.
- *Ce camion très lourd* roulait.
- *Mon camion bleu et ma belle voiture* auraient roulé.
- *Chaque camion* roulerait.
- *Divers camions* ont roulé.
- *Quelques gros camions et quelques voitures* avaient roulé.
- *Nul camion* ne roule.
- *Plusieurs très gros camions et plusieurs voitures* roulent.
- *N'importe quelle voiture* aurait roulé.
- *Un tel camion et une telle belle voiture* roulent.
- *Une quelconque voiture* roulât.
- *Un certain camion* roula.
- *Certaines trop grosses voitures* roulent.

***Forme 3 :** Forme subjonctive du groupe du verbe*



Notes :

- Tel que mentionné précédemment, en analyse syntaxique traditionnelle, la locution « il faut que » ne fait pas partie du groupe du sujet. Mais par simplicité et convenance, elle en fait partie pour ce projet.
- Les formes conjuguées possibles du verbe falloir sont listées au Tableau 3.23.
- Ce qu'on réfère ici au « groupe du sujet » est la liste de mots qu'on retrouve soit à la Forme 1 ou la Forme 2 plus haut. Par souci de clarté et de concision, ces deux formes n'ont pas été répétées ici.

Exemples (incluant le groupe du verbe) :

- *Il faut que le camion* roule.

- *Il aurait fallu qu'un gros camion* roulât.
- *Il faudrait que mon camion bleu et ma belle voiture* roulent.

3.5.3.2.4. Complément du nom du type « du »

On a vu à la section précédente qu'il est possible de modifier les noms communs en y apposant des adjectifs accompagnés ou non d'adverbes. Il est aussi possible de modifier ou préciser les noms communs en leur ajoutant des *compléments du nom*. On peut par exemple écrire :

- « la vitesse du camion »

Dans un tel cas, « du camion » est le complément du nom « vitesse ». On ne considère pour ce projet l'inclusion de compléments du nom que dans le cas où le groupe sujet comporte un ou des noms communs. Il va de soi qu'on ne peut écrire par exemple : « il du camion ». Il serait cependant acceptable d'écrire « celui du camion ». Mais par simplicité et convenance, seuls les groupes du sujet comportant des noms communs pourront être accompagnés de compléments du nom pour ce projet.

L'emploi de « du » dans l'exemple précédent correspond à un complément du nom qui est au masculin singulier. Les autres options suivantes sont possibles, selon le genre et le nombre du complément choisi :

- « la vitesse des camions »
- « la vitesse de la voiture »
- « la vitesse des voitures »

Il est à noter que pour le féminin singulier, on utilise deux mots, soit la préposition « de » suivie de l'article « la ». Au masculin, la combinaison équivalente « de le » n'étant pas permise en français, on doit utiliser l'article contracté « du ». De la même façon, on emploie au pluriel l'article contracté « des » au lieu de la formation incorrecte « de les ».

On constate aussi que le genre et le nombre du complément du nom ne dépend pas du tout du genre et du nombre du sujet « vitesse ». De la même façon, on aurait pu écrire :

- « les chargements du camion »
- « les chargements des camions »
- « les chargements de la voiture »
- « les chargements des voitures »

Dans le cas où deux noms communs sont utilisés dans le groupe du sujet, on permet pour ce projet d'introduire un complément du nom de type « du » pour chacun de ces noms. On permet par exemple :

- « la vitesse de la voiture et le chargement du camion »

De plus, on permet aussi l'utilisation de deux noms communs pour le complément du nom, comme dans l'exemple suivant :

- « la vitesse de la voiture et du camion »

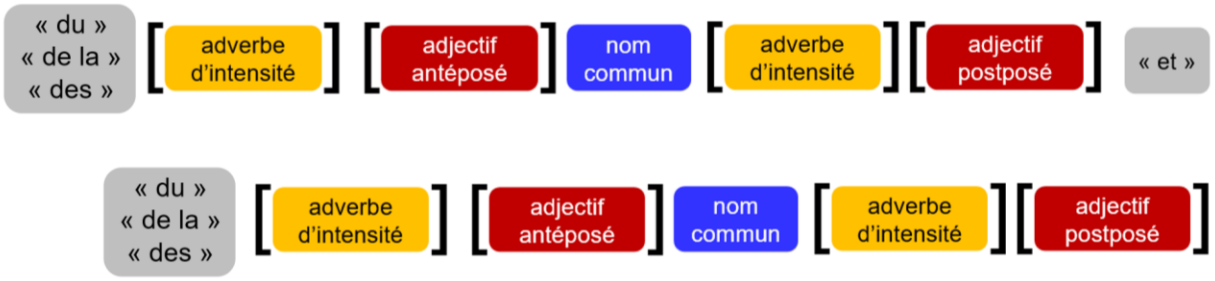
Cependant, par simplicité et convenance, bien que tout à fait convenable grammaticalement, on ne permet pas pour ce projet de complément du nom de type « du » impliquant des pronoms, comme :

- « la vitesse de celui-ci »

On peut par la suite, tout comme pour le groupe du nom du groupe sujet, ajouter arbitrairement un adjectif et un adverbe d'intensité, comme pour l'exemple suivant :

- « la vitesse du très gros camion »

La formation d'un complément du nom qu'on qualifie ici de type « du » est donc simple. En effet, le seul accord à considérer est entre les mots « du », « de la » ou « des », et le groupe du nom suivant. La séquence des mots du complément du nom de type « du » est affichée plus bas :



3.5.3.2.5. Complément du nom du type « qui »

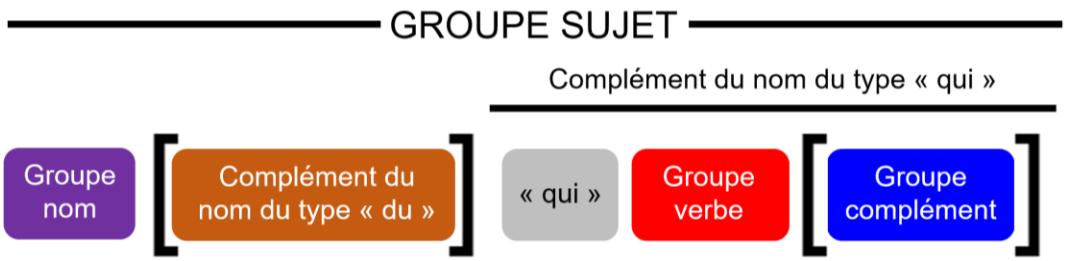
Le sujet du groupe du sujet, peut aussi être modifié en y ajoutant un complément du nom qu'on définit ici comme étant du type « qui ». Dans un tel cas, on assigne au sujet une action additionnelle le distinguant, avant de passer au groupe du verbe principal. On pourrait par exemple écrire :

- « La voiture *qui roule vite*, vient de passer sur la lumière rouge. »

Dans cette phrase, les mots « qui roule vite » forment le complément du nom du type « qui ». On constate que le complément du nom de type « qui » se distingue du complément du nom du type « du » à certains égards. En effet, le complément du nom de type « du » ne contient que l'équivalent de la préposition « de » suivie d'un groupe du nom. Tandis que le complément du nom de type « qui » ne contient *pas* de groupe du nom. Il contient plutôt un groupe verbe, et dépendamment du verbe choisi, un groupe complément. On pourrait en effet écrire :

- « La voiture *qui roule sur le chemin*, vient de passer sur la lumière rouge »

Cette dernière forme comprend ainsi le complément d'objet indirect « sur le chemin ». Sans aller dans le détail de chaque mot le comprenant, on illustre plus bas l'ordre des groupes de mots en présence d'un complément du nom de type « qui » au sein du groupe sujet.



On constate ici que même en présence d'un complément du nom du type « qui », on permet la présence d'un complément du nom de type « du ». Ainsi, on pourrait générer la phrase :

- « La voiture du garage qui roule sur le chemin (…) »

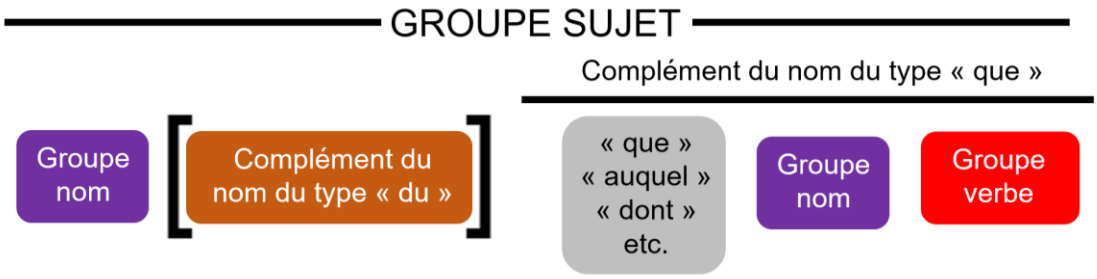
La présence d'un groupe complément au sein du complément du nom de type « qui » dépend du verbe utilisé dans le groupe verbe du complément. Davantage de détails seront fournis concernant les compléments et les liens qui les associent avec les verbes, à la Section 3.5.3.3.

3.5.3.2.6. Complément du nom du type « que »

Par souci de simplicité, on définit ici les compléments du nom du type « que » comme englobant des compléments introduits en fait par un certain nombre de pronoms et prépositions ne se limitant pas à « que » :

- « que »
- « auquel », « auxquels », « à laquelle » et « auxquelles »
- « dont »
- « avec », « dans » et « contre » suivis de « lequel », « lesquels », « laquelle » ou « lesquelles »

On regroupe ici toutes ces possibilités au sein d'une seule catégorie car elles partagent une structure commune, qu'on illustre plus bas :



Si on compare les compléments du nom du type « que » avec ceux du type « qui » cités plus haut, on constate qu'ils contiennent un groupe nom, mais pas de groupe complément. Voici quelques exemples de compléments du nom du type « que » impliquant certaines des formes possibles :

- « La voiture du garage *que j'ai empruntée* (…) »
- « La promesse *à laquelle le garçon a cru* (…) »
- « La personne *dont le monsieur te parle* (…) »
- « Le mur *contre lequel nous nous sommes heurtés* (…) »

Le premier de ces exemples inclut un complément du nom du type « du », pour illustrer le fait que ces deux types de compléments du nom (« du » et « que ») peuvent effectivement cohabiter dans les phrases générées au cours de ce projet. Il est à noter cependant, que par convenance et simplicité, aucun groupe sujet ne contiendra à la fois un complément du nom de type « qui » et un complément du nom de type « que », bien que grammaticalement, cela aurait été tout à fait acceptable. On constate aussi que le groupe du nom inclus dans le complément du nom du type « que » peut inclure soit des pronoms, soit des noms communs. Il n'y a aucune restriction à cet égard.

Il est à noter que la raison pour laquelle les compléments du nom du type « que » ne contiennent pas de groupe complément est que le groupe nom du groupe sujet fait office de complément pour le verbe du complément du nom. On pourrait par exemple écrire les deux variations suivantes :

- « La voiture que la femme a vue (…) »
- « La femme a vu la voiture (…) »

On constate donc à la lumière de cet exemple que le complément du nom « que la femme a vue » dans la première phrase se réfère à la voiture, qui peut donc en quelque sorte être interprétée

comme étant ici le complément du verbe « voir ». De ce constat découlent deux règles importantes :

1- Les compléments du nom de type « que » ne peuvent accepter de verbes intransitifs (verbes n'acceptant pas de compléments).
2- Dans le cas où un temps composé est utilisé dans le complément du nom, le participe passé doit s'accorder en genre et en nombre avec le groupe du nom du groupe sujet.

On peut illustrer la première règle avec l'exemple incorrect suivant :

- « La voiture que tu arrives (…) » (incorrect car « arriver » est intransitif et n'accepte donc pas de complément)

De la même façon, par simplicité et convenance, on n'acceptera pas non plus de verbes demandant une préposition suivie d'un infinitif pour les compléments du nom de type « que ». Pour ce projet en effet, on ne permet des compléments du nom du type « que » uniquement dans les cas de verbes étant transitifs directs (« T ») et transitifs indirects (« Ti ») non suivis d'un infinitif. Lorsqu'un verbe est intransitif ou lorsqu'il est suivi d'une préposition introduisant un infinitif, on fera plutôt appel à un complément du nom du type « qui ».

En ce qui concerne la deuxième règle, celle concernant l'accord du participe passé dans le complément du nom du type « que », on constate qu'il est critique de bien arrimer le complément avec le groupe sujet. On aura en effet :

- « Le camion que la femme a vu (…) »
- « Les voitures que l'homme avait achetées (…) »

On constate que le participe passé « vu » dans le premier exemple doit s'accorder en genre et en nombre avec le sujet « camion », et non avec le nom « femme » inclus au complément. De la même façon, le participe passé « achetées » dans le deuxième exemple doit s'accorder en genre et en nombre avec le sujet « voitures » et non avec le nom « homme » inclus au complément.

3.5.3.2.7. Informations fournies en sortie par l'algorithme du groupe du sujet

Une fois tous les mots du groupe du sujet déterminés, certaines informations doivent être fournies en sortie, afin d'ultimement former une phrase complète. Le Tableau 3.31 fournit la liste de tous les éléments retournés en sortie par l'algorithme du groupe du sujet. Des exemples sont fournis pour un groupe du sujet particulier déterminé arbitrairement.

La liste des mots permet de former la phrase complète, tandis que les listes de lemmes, classes grammaticales et paramètres morpho-syntaxiques sont utiles pour l'évaluation de lemmatiseurs existants (Chapitre 6). Les mots pour tableaux de cooccurrences se rajoutent aux mots issus du groupe du verbe, pour permettre de choisir les mots au hasard pour le groupe du complément. Le genre et le nombre demeurent utiles dans le cas où un verbe attributif est utilisé dans le groupe du verbe, car l'adjectif ou le participe passé suivant le verbe attributif, qu'on retrouve au groupe complément, doit s'accorder en genre et en nombre avec le sujet.

Tableau 3.31 : Éléments en sortie de l'algorithme du groupe du sujet avec exemples pour le groupe du sujet « un camion et une très grosse voiture »

Éléments en sortie	Exemples
Liste des mots (fléchis)	« un », « camion », « et », « une », « très », « grosse », « voiture »
Liste des lemmes	« un », « camion », « et », « un », « très », « gros », « voiture »
Liste des classes grammaticales	déterminant, nom commun, conjonction, déterminant, adverbe, adjectif, nom commun
Paramètres morpho-syntaxiques	un : article indéfini, masculin singulier camion : nom commun, masculin singulier et : conjonction une : article indéfini, féminin singulier très : adverbe d'intensité grosse : adjectif, féminin singulier voiture : nom commun, féminin singulier
Mots pour tableaux de cooccurrences	« camion », « très », « gros », « voiture »
Genre	masculin (en présence de masculin et féminin)
Nombre	pluriel

3.5.3.3. Génération du groupe complément

La structure du groupe du complément dépend du type du verbe utilisé au sein du groupe du verbe. Ces types sont des caractéristiques de chacun des verbes. On retrouve presque toutes ces caractéristiques dans la liste des verbes du Bescherelle, comme on peut le voir à la Figure 3.1. Ces types de verbes sont décrits au Tableau 3.32.

Pour arrimer le groupe du complément aux deux autres groupes, on doit donc en extraire les paramètres listés au Tableau 3.32. Ces paramètres, fournis en sortie par les algorithmes du groupe du verbe et du groupe du sujet, sont décrits au Tableau 3.33.

Tableau 3.32 : Types de verbe

Types de verbe	Codes utilisés dans le Bescherelle	Descriptions	Exemples
Attributif	Aucun	Ces verbes se construisent avec un attribut du sujet. Les verbes attributifs utilisés pour ce projet sont : demeurer, devenir, être, mourir, naître, paraître, rester, sembler, tomber	Il *reste* fâché. Je *deviens* vieux.
Intransitif	I	Ces verbes n'acceptent aucun complément	Le garçon *bavarde*. Elles *bafouillent*.
Transitif direct	T	Ces verbes acceptent un complément d'objet direct	L'homme *mange* une pomme.
Transitif indirect sans infinitif	Ti	Ces verbes acceptent un complément d'objet indirect demandant l'usage d'une préposition. Des prépositions précises sont associées à chaque verbe	La montagne a *accouché* d'une souris. J'*adhère* à ses idées.
Transitif indirect avec infinitif	Ti + inf	Ces verbes introduisent un infinitif en conjonction avec l'usage d'une préposition donnée.	Elle a *cessé* de respirer. Il l'*aide* à se relever.
Pronominal	P	Ces verbes demandent l'utilisation d'un pronom réfléchi. Sans conséquence pour le groupe du complément	Je me *lave*.

Tableau 3.33 : Éléments en sortie de l'algorithme du groupe du verbe et du groupe du sujet requis pour la génération du groupe du complément

Éléments en sortie	Groupe dont l'élément est issu	Utilité pour le groupe du complément
Type du verbe	Groupe du verbe	Le type du verbe (attributif, intransitif, transitif direct, transitif indirect avec ou sans infinitif) détermine le type de complément nécessaire
Préposition	Groupe du verbe	Dans le cas de verbes transitifs indirects, une préposition est fournie par le groupe du verbe, pour inclure au complément
Nombre	Groupe du sujet	En cas de verbe attributif, l'attribut doit s'accorder en nombre avec le groupe du sujet
Genre	Groupe du sujet	En cas de verbe attributif, l'attribut doit s'accorder en genre avec le groupe du sujet
Mots pour tableaux de cooccurrences	Groupes du verbe et du sujet	On détermine au hasard les lemmes à employer dans le groupe complément parmi les cooccurrences des mots utilisés jusqu'à présent dans les groupes du verbe et du sujet

En plus des éléments inclus au Tableau 3.32, le Tableau 3.33 comporte les mots pour les cooccurrences. En effet, toujours dans le but de donner un minimum de sens aux phrases générées, on tente, pour le groupe du complément, de choisir des mots qu'on retrouve dans le corpus de référence dans les mêmes phrases que les mots utilisés jusqu'à présent dans les deux autres groupes de la phrase. Les prochaines sections décrivent comment le groupe du complément est bâti, selon le type de verbe inclus au groupe du verbe.

3.5.3.3.1. Cas avec verbe intransitif

Selon le Bescherelle (2012), « certains verbes expriment à eux seuls le procès complet et peuvent se passer d'autres compléments ». En général, ces verbes expriment un état ou une action. Ils expriment une action limitée au sujet et ne passant sur aucun objet. Ils ne sont donc jamais suivis d'un complément d'objet. Dans le contexte de la génération de phrases aléatoires, le cas intransitif est donc le plus simple, car ces verbes ne requièrent aucun groupe du complément. Le groupe du complément demeure donc vide et la phrase se termine avec le groupe du verbe. Deux exemples sont fournis au Tableau 3.32.

3.5.3.3.2. Cas avec verbe attributif

Un verbe attributif exprime l'état du sujet, plutôt que de décrire son action. Dans le cadre de ce projet, l'attribut est toujours un adjectif ou un participe passé, dont le genre et le nombre s'accordent donc avec le groupe du sujet. La liste de verbes attributifs utilisés pour ce projet, de même que certains exemples sont fournis au Tableau 3.32. Certains verbes sont considérés « essentiellement attributifs ». Ceux-ci doivent obligatoirement être accompagnés d'un attribut. Par exemple, on ne peut écrire « Je deviens. » D'autres verbes sont occasionnellement attributifs, c'est-à-dire que dépendamment du contexte, ils demandent un attribut du sujet ou un autre type de complément. Dans le cadre de ce projet, par simplicité, tous les verbes pouvant être attributifs sont traités comme tel et sont donc toujours suivis d'un adjectif ou d'un participe passé. L'adjectif ou participe passé est choisi en priorité parmi les cooccurrences combinées de tous les lemmes générés dans les groupes du verbe et du sujet. L'algorithme s'assure d'en accorder le genre et le nombre avec le groupe du sujet. Pour ce projet, on détermine par la suite avec une probabilité arbitraire, si l'adjectif ou participe passé est précédé ou non d'un adverbe d'intensité. Ainsi on peut avoir « Il devient fâché » ou « Il devient *très* fâché ».

3.5.3.3.3. Cas avec verbe transitif direct

Comme le décrit le Bescherelle (2012), le verbe transitif « est employé avec un complément d'objet, sur lequel s'exerce ou passe l'action du sujet exprimée par le verbe. Lorsque le complément n'est pas précédé par une préposition, il est dit direct. » Dans un tel cas, le groupe du complément se compose d'un « groupe du nom ». Ce groupe du nom est presque en tout point semblable à ce qu'on génère au groupe du sujet. En effet, pratiquement le même algorithme utilisé pour le groupe du sujet est aussi utilisé pour générer le groupe du nom pour les compléments d'objet direct.

En effet, pour ce projet, le groupe du complément dans le cas transitif peut contenir soit un ou des noms communs, soit des pronoms. Les noms communs peuvent être accompagnés d'adjectifs et d'adverbes. On peut aussi inclure des compléments du nom tels que décrits aux Sections 3.5.3.2.4 à 3.5.3.2.6. Les mêmes règles décrites pour le groupe du sujet sont ainsi appliquées pour le complément d'un verbe transitif.

Pour les compléments impliquant des noms communs, la séquence des mots en sortie est exactement la même que pour le groupe du sujet (Forme 2 de la Section 3.5.3.2.3). Voici quelques exemples (incluant les groupes du sujet et du verbe) :

- Le garçon mange *la pomme*.
- Le garçon mange *une grosse pomme*.
- Le garçon mange *cette pomme verte*.
- Le garçon mange *cette pomme et ce bleuet*.
- Le garçon mange *chaque pomme*.
- Le garçon mange *diverses pommes*.
- Le garçon mange *quelques pommes jaunes et quelques bleuets*.
- Le garçon ne mange *nulle pomme*. (« nulle » utilisé uniquement sous la forme négative)
- Le garçon mange *plusieurs pommes et plusieurs gros bleuets*.
- Le garçon mange *n'importe quelle pomme*.
- Le garçon mange *une telle pomme verte et un tel bleuet*.
- Le garçon mange *une quelconque pomme*.

Voici maintenant quelques exemples similaires incorporant des compléments du nom au sein du groupe du complément :

- Le garçon mange *la pomme du verger*. (complément du nom du type « du »)
- Le garçon mange *la pomme dont sa mère parlait*. (complément du nom de type « que »)
- Le garçon mange *la pomme qui gisait devant lui*. (complément du nom de type « qui »)

Tous les exemples précédents ont utilisé un nom commun (« pomme ») pour former le complément d'objet direct. On a constaté que le groupe du nom qui en a résulté était en tout point semblable à celui bâti pour le groupe du sujet. Cependant, lorsqu'on utilise un pronom plutôt qu'un nom commun, le complément d'un verbe transitif direct se bâtit différemment que dans le cas du groupe du sujet. En effet, lorsque le complément d'objet direct est un pronom, celui-ci doit être déplacé *devant* le verbe. Par exemple, tandis qu'on écrit « elle mange un bleuet », on doit écrire « elle *le* mange » en parlant de ce même bleuet. On constate que le pronom « le » a dû être déplacé devant le verbe.

En présence de pronoms, l'algorithme du groupe du complément doit donc être en mesure de placer ceux-ci au bon endroit. À strictement parler, le pronom « le » dans la phrase « Elle le mange » ne fait pas partie du groupe du verbe tel qu'il a été défini pour ce projet à la Section 3.5.2. Mais par convenance, c'est bien ici dans le groupe du verbe que l'on déplace ces pronoms compléments d'objet direct pour ce projet. Ainsi, l'algorithme de création de phrases doit modifier le groupe du verbe en conséquence, pour y insérer le pronom.

Le positionnement du pronom objet au sein du groupe du verbe n'est pas trivial. En effet, sa position dépend de plusieurs facteurs, incluant le temps du verbe, l'utilisation de verbe modal, l'utilisation de la forme affirmative ou négative, et l'utilisation de temps composés avec le participe passé. Toutes les possibilités de positionnement du pronom sont présentées ici :

- Dans le cas de l'impératif, le pronom objet est situé après le verbe : « Mange-*le* ! ». En présence d'adverbes, le pronom s'insère devant eux : « Mange-*le* très lentement ! ».
- Dans un cas non modal et affirmatif, on insère le pronom devant le verbe : « Je *le* mange ». En présence d'adverbes, le pronom s'insère devant eux : « Je *le* mange très lentement ».
- Dans un cas non modal et négatif, on insère le pronom devant le verbe et après l'adverbe « ne » : « Je ne *le* mange pas ». En présence d'adverbes, le pronom s'insère devant eux : « Je ne *le* mange pas très lentement ».

- Dans un cas modal, affirmatif ou négatif, on insère le pronom devant l'infinitif « Je veux *le* manger » ou « Je ne veux pas *le* manger ». En présence d'adverbes, cette position demeure la même : « Je ne veux pas *le* manger très lentement ».
- Pour les temps composés avec l'auxiliaire « avoir », les règles précédentes demeurent les mêmes, mais le verbe auquel on fait référence plus haut devient l'auxiliaire « avoir ». Ainsi :
    - Dans un cas non modal et affirmatif, on insère le pronom devant l'auxiliaire « avoir »: « Je *l'*ai mangé ». En présence d'adverbes, le pronom s'insère devant eux : « Je *l'*ai mangé très lentement ».
    - Dans un cas non modal et négatif, on insère le pronom devant l'auxiliaire « avoir » et après l'adverbe « ne » : « Je ne *l'*ai pas mangé ». En présence d'adverbes, le pronom s'insère devant eux : « Je ne *l'*ai pas mangé très lentement ».
    - Dans un cas modal, affirmatif ou négatif, on insère le pronom devant l'infinitif « J'ai voulu *le* manger » ou « Je n'ai pas voulu *le* manger ». En présence d'adverbes, cette position demeure la même : « Je n'ai pas voulu *le* manger très lentement ».

Toutefois, avec l'auxiliaire « avoir », il faut se rappeler de la règle d'accord du participe passé. En effet, le participe passé conjugué avec l'auxiliaire « avoir » s'accorde en genre et en nombre avec le complément d'objet direct *si celui-ci est situé devant le verbe*. Et c'est exactement la situation à laquelle nous faisons face ici. Donc, si on fait référence à une pomme (féminin singulier) plutôt qu'à un bleuet (masculin singulier), il faudra modifier le participe passé en conséquence :

- Je l'ai mangée très lentement.
- Je ne l'ai pas mangée très lentement.

Dans le cas modal, le problème ne se pose pas, puisque le complément d'objet direct est en lien avec le verbe à l'infinitif « manger », et non avec l'auxiliaire avoir conjugué. On obtient donc :

- J'ai voulu la manger très lentement.
- Je n'ai pas voulu la manger très lentement.

Il n'y a donc dans ce cas aucun changement à apporter au participe passé « voulu ». On se contente d'utiliser le pronom approprié (ici, « la » plutôt que « le », puisqu'au féminin).

On est donc à même de constater que la formation du groupe du complément dans le cas de verbes transitifs directs avec des pronoms requiert des modifications au groupe du verbe, ce qui complique l'algorithme de création de phrases.

3.5.3.3.4. Cas avec verbe transitif indirect non-accompagné de l'infinitif

Un verbe est transitif indirect lorsque le complément d'objet qu'il introduit exige la présence d'une préposition, elle-même suivie soit d'un infinitif, soit d'un groupe du nom. Dans la section présente, il est question des cas impliquant un groupe du nom. Le groupe du nom pouvant être composé ou bien de noms communs ou bien de pronoms, il y a donc deux cas à considérer.

Dans le cas du groupe du nom avec noms communs, on forme le groupe du complément de la même façon que pour le groupe du sujet. Il faut cependant insérer la préposition devant le groupe du nom :

- Le garçon goûte *à* la très grosse pomme.

Quand deux noms communs sont utilisés, il faut inclure la préposition devant chacun des noms :

- Le garçon goûte *à* la très grosse pomme et *à* la poire.

Là encore, les noms communs choisis peuvent être accompagnés d'adjectifs et d'adverbes, dont la présence est déterminée aléatoirement de la même façon qu'on l'a démontré pour le groupe du sujet.

Le cas du complément d'objet indirect avec pronoms est bien différent de l'objet direct, tel que vu à la section précédente. Pour l'objet direct, on a vu qu'il faut déplacer le pronom au sein du groupe du verbe. Tel n'est pas nécessairement le cas pour les compléments d'objet indirect. De plus, il faut être prudent dans le choix des pronoms à employer. Considérons par exemple la phrase « Elle parle à son ami ». Si on veut remplacer « son ami » par un pronom, deux options sont possibles :

- « Elle lui parle. »
- « Elle parle à lui. »

La première option est la plus « correcte », mais la seconde demeure grammaticalement acceptable. Dans le cadre de ce projet, par simplicité et convenance, c'est la deuxième option qui est adoptée, car elle ne nécessite pas de déplacement du pronom. Cependant, il faut utiliser ici le type de pronom personnel approprié, soit du type « datif », tel que discuté au Tableau 3.7. On peut ainsi écrire « Elle parle à moi », « Elle parle à toi », « Elle parle à nous », etc.

Des pronoms autres que les pronoms personnels peuvent aussi être employés, de la même façon que pour le groupe du sujet. On peut écrire par exemple :

- « Elle parle à la mienne. »
- « Elle parle à toutes. »
- « Elle parle à celui-ci. »
- « Elle ne parle à personne. » (« personne » est utilisé exclusivement pour la forme négative)
- « Elle parle à quelqu'un. »
- « Elle parle à quiconque. »

La séquence des mots pour le groupe du complément avec un verbe transitif indirect (sans infinitif) est donc la même que pour le groupe du sujet, à la différence que les prépositions doivent être insérées aux bons endroits.

3.5.3.3.5. Cas avec verbe transitif indirect accompagné de l'infinitif

Certains verbes s'accompagnent d'une préposition ensuite directement suivie d'un infinitif. On a par exemple :

- « Il cesse *de* respirer. »
- « Il m'aide *à* manger. »
- « Elle consent *à* venir. »
- « Ils tentent *de* s'approcher. »

Ces verbes sont identifiés dans la liste de verbes du Bescherelle avec le code « Ti+inf ». Construire le groupe du complément dans ce cas implique donc d'insérer en premier lieu la préposition appropriée, et de choisir ensuite aléatoirement un second infinitif. Une fois ce deuxième infinitif incorporé, on peut alors poursuivre avec un groupe du complément plus détaillé. On pourrait par exemple se retrouver avec les formes suivantes :

- « Il cesse de respirer l'air pur. »
- « Il m'aide à manger mon repas. »
- « Elle consent à venir me chercher. »
- « Ils tentent de s'approcher de la porte. »

On constate qu'on peut ensuite enchainer avec un complément d'objet direct (« l'air pur », « mon repas »), ou encore avec un autre verbe qui peut alors être transitif ou non (« me chercher »), ou encore un complément d'objet indirect (« de la porte »). Mais dans le cadre de ce projet, par simplicité, les formes permises suivant l'infinitif sont limitées. On ne permet pas de pronominal, ni de verbe modal, et ni d'autre infinitif. En particulier, si on permettait d'introduire un nouvel infinitif, on risquerait de se retrouver dans une boucle infinie, telle que « elle commence à commencer à commencer à commencer… ». Les autres possibilités auraient pu être considérées, mais auraient mené à des phrases plus complexes. De cette simplification résulte que l'algorithme actuel ne permet pas de générer la phrase « elle consent à venir me chercher ».

L'algorithme pour le groupe du complément, lorsqu'un transitif indirect avec infinitif est imposé, sélectionne d'abord aléatoirement l'infinitif puis l'adjoint à la préposition fournie par le groupe du verbe. Le reste du groupe du complément se détermine ensuite de façon aléatoire en permettant diverses formes, incluant un complément d'objet direct ou indirect.

3.5.4. Résumé des règles d'accords entre les différents groupes de la phrase

Tel que mentionné à la Section 3.2.3.2, les règles grammaticales du français font en sorte qu'on doit par exemple accorder les déterminants avec les noms qu'ils déterminent ainsi que les adjectifs avec les noms qu'ils modifient. Ces accords grammaticaux s'exécutent généralement au sein d'un même groupe de la phrase (sujet ou complément). Mais en plus de ces règles d'accord grammatical s'effectuant au sein d'un groupe donné (intra-groupe), on a vu que d'autres règles grammaticales imposent des accords entre les différents groupes (extra-groupe). Entre autres, la personne du verbe dans le groupe du verbe influe sur le nombre du sujet, et même parfois sur son genre, dans le cas de temps composés. On a aussi vu que les verbes attributifs forcent l'adjectif ou participe passé du complément (l'attribut) à s'accorder en genre et en nombre avec le groupe du nom du groupe sujet.

La Figure 3.6 résume toutes les règles d'accord grammatical s'appliquant d'un groupe de la phrase à un autre (ignorant les règles s'appliquant au sein de chaque groupe). Cette figure permet aussi d'illustrer les différentes structures de phrases qui pourront être générées par l'algorithme de création de phrases développé pour ce projet. Il est toutefois bon de rappeler que bon nombre des structures apparaissant à la Figure 3.6 sont optionnelles.

Cette figure nous indique par exemple que le nombre du groupe du nom du sujet (indiqué ici par « Groupe Nom 1 ») doit s'accorder avec le nombre du « Groupe Verbe 1 », selon que la personne du verbe est au pluriel ou au singulier. Cette relation grammaticale est illustrée ici par l'expression « *n = GV1.n* », qui signifie que le nombre (« *n* ») est égal au nombre du « Groupe Verbe 1 ». La nomenclature adoptée ici a été inspirée de la notation employée dans les langages de programmation orientés objet. De la même façon, on constate que si le « Groupe Verbe 1 » est un attributif, l'attributif (case bleu pâle en haut à droite) doit s'accorder en genre et en nombre avec le « Groupe Nom 1 », tel qu'explicité par les expressions « *n = GN1.n* » et « *g = GN1.g* » ou « *g* » représente le genre. Il est à noter que dans les expressions « *n = -1* » et « *g = -1* », la valeur « *-1* » indique que les nombres et genres sont dans ces cas arbitraires, une convention utilisée lors de la programmation de ces algorithmes en langage Java, tel que discuté au Chapitre 4.

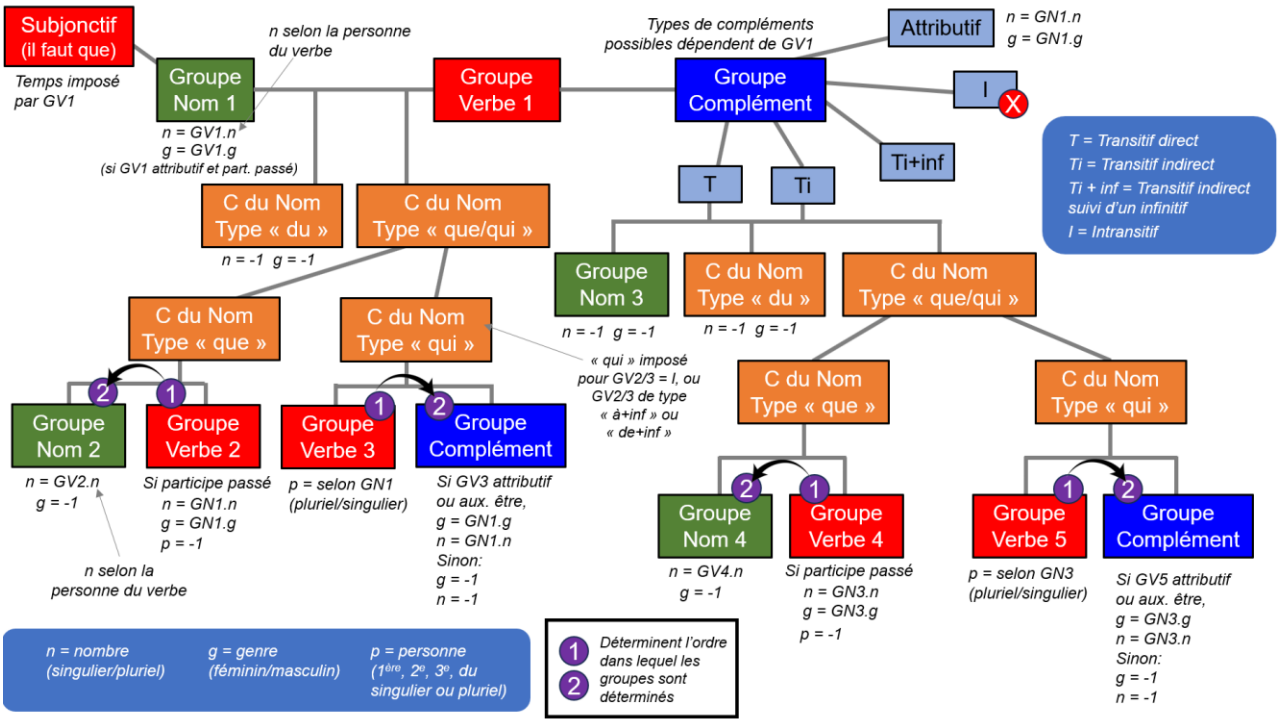


Figure 3.6 : Vue d'ensemble de l'application des règles d'accord grammatical entre les différents groupes de la phrase

La Figure 3.6 met ainsi en relief certaines règles d'accord non triviales entre les composants des compléments du nom et les groupes environnants. Finalement, pour les compléments du nom, les cercles comprenant les nombres « 1 » et « 2 » servent à indiquer l'ordre dans lequel les groupes sont générés par l'algorithme. La Figure 3.6 donne donc un aperçu d'ensemble du fonctionnement de l'algorithme de création de phrases développé pour ce projet.

3.5.5. Formation des phrases complètes

Une fois les trois groupes complétés (groupe du verbe, groupe du sujet et groupe du complément), tous les ingrédients sont en place pour générer la phrase complète. En effet, les phrases générées aléatoirement pour ce projet se limitent aux structures relativement simples illustrées à la Figure 3.6. Cette dernière étape consiste en fait à combiner l'information générée par les algorithmes associés à ces trois groupes. On met donc bout à bout les informations en lien avec les mots, lemmes, et formes morpho-syntaxiques de chaque groupe pour se retrouver avec un ensemble unique contenant ces informations pour la phrase dans son entièreté.

On doit ensuite incorporer des éléments essentiels pour former la phrase. Tout d'abord, une phrase débute toujours par une majuscule. La première lettre du premier mot inclus dans le groupe du sujet sera donc une majuscule. Les phrases se terminent aussi par un point, sauf pour les phrases impératives que l'algorithme fait terminer avec un point d'exclamation. Il est à noter qu'aucune phrase interrogative n'est générée par l'algorithme développé ici.

Des virgules sont aussi insérées pour séparer les compléments du nom de type « qui » et de type « que » tel que dans les exemples suivants :

- « Le garçon, qui mange de la crème glacée, conduit une voiture. »
- « La jeune fille, que ma mère connaît, se dirige vers l'école. »

En revanche, aucune virgule n'est insérée dans le cas de compléments du nom de type « du », comme dans les exemples suivants :

- « Le fils de ma sœur conduit une voiture. »
- « La mère de mon ami, qui aime manger, fait de la bicyclette. »

En effet, pour le deuxième exemple, des virgules sont insérées pour délimiter le complément du nom de type « qui » (« qui aime manger »), mais pas pour le complément du nom de type « du » (« de mon ami »).

Après les points et les virgules, il faut ensuite considérer un autre type de ponctuation : les apostrophes. En effet, plusieurs mots laissent tomber leur dernière voyelle (élision) et requièrent une apostrophe lorsque le mot suivant débute par une voyelle ou un « h » muet. Une liste non exhaustive de tels mots est fournie au Tableau 3.34.

Pour le cas des mots débutant par lettre « h », il faut être prudent. Seulement certains d'entre eux impliquent l'élision (utilisation de l'apostrophe). Par exemple, on écrit « l'heure » et non « la heure ». Au contraire, on écrit « le handicapé », et non « l'handicapé ». L'algorithme actuel fait appel à une banque de données de mots débutant par « h » demandant l'élision. Pour ces derniers, les mêmes règles s'appliquent pour l'introduction de l'apostrophe que pour les mots débutant par une voyelle.

Cette étape d'introduction des apostrophes s'effectue de façon optimale une fois que la phrase complète est formée, puisque certaines apostrophes peuvent se retrouver nécessaires au moment d'arrimer entre eux les trois groupes. Un algorithme, décrit plus en détail au Chapitre 4, est donc mis en place pour insérer les apostrophes partout où cela est requis. Compte tenu de la simplicité relative des phrases générées dans ce projet, aucune autre marque de ponctuation (point-virgule, deux-points, guillemets) n'est nécessaire.

Un dernier ajustement est requis, celui de la formation potentielle d'articles contractés. Si par exemple un verbe doit être suivi de la préposition « de », et que le groupe du complément est composé par la suite des mots « le camion », il est obligatoire de combiner les mots « de » et « le » pour former l'article contracté « du », pour finalement obtenir « du camion » plutôt que « de le camion ». Il en est de même pour la préposition « à » suivi du déterminant « les », pour former l'article contracté « aux ». Bien qu'un lemmatiseur fonctionnerait surement aussi bien en l'absence d'apostrophes (« la automobile ») ou en présence d'articles « non contractés » (« de le camion »), il est essentiel d'effectuer ces ajustements pour générer un texte français bien construit.

Finalement, l'information fournie en sortie pour chaque phrase complète est décrite au Tableau 3.35. L'ensemble de ces phrases compose donc le texte aléatoire automatiquement lemmatisé, objectif principal de la Section 3.5.3. Un tel texte servira d'étalon, au Chapitre 6, pour l'évaluation d'outils de lemmatisation existants.

Tableau 3.34 : Liste non exhaustive de mots subissant l'élision devant un autre mot commençant par une voyelle ou un « h » muet

Classe grammaticale	Mots subissant l'élision
Déterminants	le, la, l', de, ce, que, quelque
Pronoms	je, me, te, se
Adverbes	ne
Conjonctions	si (lorsque suivi des pronoms sujet « il » ou « ils »), lorsque, puisque, quoique
Prépositions	jusque

Tableau 3.35 : Information en sortie pour chaque phrase aléatoire automatiquement lemmatisée, avec l'exemple d'une phrase arbitraire (« Les grands amis voyageaient ensemble. »)

Éléments en sortie	Exemples
Liste des mots (fléchis)	les, grands, amis, voyageaient, ensemble
Liste des lemmes	le, grand, ami, voyager, ensemble
Liste des classes grammaticales	5, 2, 3, 1, 4
Paramètres morpho-syntaxiques	(51, 0, 1), (2, 0, 1), (31, 0, 1), (1, 2, 6), (4)
Nombre de mots dans la phrase	5

3.5.6. Salade de mots

Une lemmatisation complète et efficace d'un texte requiert dans le cas général, la désambiguïsation des potentiellement nombreux homographes en faisant partie. Tel que discuté à la Section 3.2.3, une stratégie efficace consiste à procéder à l'analyse syntaxique de la phrase, afin d'éliminer certaines possibilités d'homographes en fonction de leur classe grammaticale. Une telle analyse syntaxique requiert une analyse des autres mots composant la phrase, et particulièrement la séquence dans laquelle ils apparaissent.

Mais il est possible de concevoir un lemmatiseur « de base », ne tenant pas du tout en compte la syntaxe des phrases. Par exemple, un tel lemmatiseur pourrait, à chaque fois qu'il croise un homographe, choisir celui correspondant à la classe grammaticale la plus fréquente dans la langue française, parmi celles identifiées comme possibles pour cet homographe. Ou encore, le lemmatiseur pourrait utiliser des statistiques globales sur le corpus de référence en entier, et évaluer, par exemple, la possibilité que le mot « plus » soit une forme du verbe « plaire » plutôt que l'adverbe « plus » sur la base de la présence du verbe plaire à d'autres temps et personnes dans le texte. Il est fort à parier qu'un tel lemmatiseur n'atteindrait pas l'efficacité des meilleurs lemmatiseurs disponibles actuellement, qui permettent dans certains cas un taux de succès pour la désambiguïsation de plus de 97% pour le français (Bourdaillet & Ganascia, 2005). Tout de même, de tels outils peuvent s'avérer pratiques pour des situations où la précision n'est pas critique, et pour lesquelles un outil simple, rapide et peu couteux convient mieux.

Le projet actuel offre un outil permettant justement de déterminer si un outil de lemmatisation prend en compte la syntaxe des phrases. Pour y arriver, un algorithme a été développé pour créer une « salade de mots » (« *bag of words* » en anglais), qui consiste à prendre tous les mots de toutes les phrases générées à l'étape précédente, et à les recombiner de façon complètement aléatoire. Ainsi, la « salade de mots » qui en résulte se retrouve avec exactement les mêmes mots, les mêmes lemmes et les mêmes statistiques globales que le texte aléatoire original, mais listés dans un tout autre ordre, où la syntaxe n'est plus de mise.

Cet algorithme de salade de mots forme tout de même des phrases « artificielles » dont le nombre de mots correspond à celui des phrases générées à l'étape précédente. Ainsi on crée pour cette salade de mots des « phrases » débutant par une majuscule et se terminant par un point, mais composées de mots dans une séquence tout à fait arbitraire. On pourrait se retrouver par exemple avec des phrases comme la suivante, complètement dénuée de sens et de syntaxe :

- « Théorie mon avions inspiré bien portons qu'électrode les fallait quasi pas. Avons sourire ne. Voulons les pas. Eusse nous demandées la. »

Un lemmatiseur ne tenant pas compte de la syntaxe et de la séquence des mots fournira donc en sortie les mêmes résultats et statistiques en analysant le texte original et sa salade de mots

correspondante. Une telle performance identique offre donc un indice indéniable que le lemmatiseur sous évaluation ne tire pas profit de la syntaxe pour faciliter la désambiguïsation. Au contraire, un lemmatiseur plus avancé appliquant l'analyse syntaxique pour la désambiguïsation obtiendra un bien meilleur résultat avec le texte original bien bâti, qu'avec la salade de mots correspondante.

L'algorithme pour générer la salade de mots est présenté en détail à la Section 4.10.8.

4. ALGORITHMES ET PROGRAMMATION

La méthodologie générale a été décrite au Chapitre 3, sans égard à comment celle-ci sera mise en application. Au chapitre présent, on effectue un retour sur la méthodologie, mais cette fois dans le contexte de la programmation des algorithmes informatiques. Pour ce projet, le langage Java (Oracle Corporation, Version 14.0.1, date de sortie 2020-04-14) a été utilisé et l'IDE Netbeans (Apache, Version 12.1, date de sortie 2020-09-05) a facilité l'usage de ce langage et l'exécution des scripts. L'objectif de ce chapitre n'est pas de détailler ligne par ligne tout le code Java développé, mais plutôt de mettre en lumière les structures de données principales adoptées et un aperçu des algorithmes dans le contexte du langage Java, pour atteindre les buts recherchés.

4.1. Création de fichiers de banques de données et algorithmes pour les traiter

Avant même de programmer, il faut créer des fichiers texte comprenant des banques de mots. Pour les verbes, les tableaux de verbes et les modèles de conjugaison contenus dans le Bescherelle (2012) fournissent l'information de base pour générer toutes les formes fléchies (conjuguées). Pour les noms communs, adjectifs et adverbes, extraire toutes les possibilités du dictionnaire aurait représenté un trop lourd travail. Ces mots ont donc été extraits du corpus de référence, un travail manuel ardu, mais ne se comparant pas à l'entrée de la majorité des mots du dictionnaire. Cette banque de mots peut être augmentée au besoin, par exemple quand des textes additionnels sont ajoutés au corpus de référence, ou quand un autre corpus est utilisé. Pour les autres classes grammaticales, des banques de mots souvent exhaustives, parfois non, disponibles sur le web ou ailleurs, ont été utilisées pour recenser les formes requises.

Comme le nombre de mots fléchis à générer est de l'ordre de plusieurs centaines de milliers (comme on le verra au Chapitre 5), il était important d'adopter une structure de données facilitant l'entrée de ces données au moment de la création des banques de mots. Mais surtout, il fallait une structure offrant un accès rapide lors de la lemmatisation, pour associer chaque mot du corpus à un mot des banques lexicales.

On a donc opté pour le concept de table de hachage comme structure de choix pour emmagasiner ces données. Tel qu'on le décrit dans les notes du cours INF1425 de la TELUQ (2023), « une table de hachage est une structure de données qui permet une association clé-élément : l'accès à un élément de la structure de données se fait en transformant la clé en une valeur de hachage par l'intermédiaire d'une fonction de hachage. » La fonction « *Hashmap* » de Java sera donc utilisée. Le fonctionnement de cette structure de données n'est pas détaillé ici, car il demeure invisible au programmeur. Ce qu'il compte de savoir est que d'une part, la table de hachage permet de facilement ajouter des données, donc des mots, sans avoir à se soucier de la séquence dans laquelle ils sont entrés. Mais surtout, l'accès aux données d'un mot précis s'effectue rapidement selon une complexité algorithmique d'ordre $\Theta(1)$. Ceci correspond à une complexité dite « constante », c'est-à-dire que son temps d'exécution ne croît pas avec la taille de la table. Cette caractéristique est essentielle dans notre cas, vu l'énorme quantité de mots inclus dans la table (quelques centaines de milliers).

On accède à chaque élément d'une table de hachage par sa clé unique, caractéristique première des tables de hachage. À cette clé est associée une « valeur » unique. On ne peut en effet associer la clé à plus d'une valeur. La valeur peut cependant prendre plusieurs formes : une variable simple (entier, flottant, caractère), ou encore un objet qui peut être une chaine de caractère ou tout autre objet plus complexe né d'une classe. Dans le contexte de ce projet, un objet issu d'une classe Java a été créé sur mesure pour emmagasiner toute l'information à associer à chaque mot, donc à chaque clé.

Mais comme certains mots sont associés à plus d'un lemme et/ou plus d'une classe grammaticale (homographes), il est essentiel que l'objet de la table de hachage permette plusieurs possibilités de lemmes. L'objet créé ici est donc composé d'une liste chaînée (« *LinkedList* »). Toujours selon les notes du cours INF 1425 de la TELUQ (2023), une liste chaînée est « une structure de données dans laquelle les objets sont rangés linéairement » et dont l'ordre « est déterminé par un pointeur sur chaque objet. » L'accès à ces objets se fait de façon séquentielle, ce qui est approprié ici, considérant le peu d'éléments dont on aura besoin. En effet, quand il s'agit d'homographes, la liste chainée contient autant d'éléments que de possibilités d'homographes, un nombre limité d'éléments. Par exemple, puisque le mot « fort » peut être ou bien un nom commun, ou bien un adverbe ou un adjectif, sa liste chainée comprend trois éléments. Au contraire, le mot « lentement », qui n'est pas un homographe, n'est associé qu'à un seul élément de liste chaînée.

Tel que mentionné plus haut, chaque élément de la liste chainée est un « objet ». Pour ce projet, l'objet en question est associé à une classe créée expressément, appelée « *LemmeObjet* ». Cette classe apparaît à l'extrait de code 4.1.

Extrait de code 4.1 : La classe « *LemmeObjet* »

```
public class LemmeObjet {

    private String mot;
    private String lemme;
    private int POS;
    private int[] param;

// ---------- METHODE SET -------------------

public void Set (String motSet, String lemmeSet, int POSSet, int [] paramSet){

mot=motSet;
lemme=lemmeSet;
POS=POSSet;
param=paramSet;
}

// ---------- METHODES GET -------------------

public String GetMot(){
return mot;
}
public String GetLemme(){
return lemme;
}
public int GetPOS(){
return POS;
}
public int[] GetParam(){
return param;
}
}
```

Chaque instance de cette classe (objet) contient le mot lui-même (String : *mot*) son lemme (String : *lemme*), sa classe grammaticale (entier de 1 à 9 : *POS*[4]), ainsi qu'une liste de paramètres (vecteur d'entiers : *param*). La signification de chaque élément du vecteur *param* dépend de la classe grammaticale, tel qu'illustré au Tableau 4.1. Pour les temps de verbes, il est question d'un entier entre 0 et 20, pour les personnes de verbes, un entier entre 1 et 6, pour le genre et le nombre, un entier valant 0 ou 1, et finalement pour le cas, un entier entre 0 et 2. Davantage de détails concernant ces paramètres ont été discutés à la Section 3.1.1.

Tous les mots du lexique, peu importe leur classe grammaticale, sont donc emmagasinés dans une unique table de hachage de type « *Hashmap* », baptisée « *tableRef* » pour ce projet. La clé pour cette table est le mot fléchi lui-même. La valeur associée à chaque clé est une liste chaînée comportant au moins un élément, et chaque élément de la liste chaînée est un objet de la classe « *LemmeObjet* », dont les variables principales sont le lemme associé à chaque mot, sa classe grammaticale, ainsi que ses paramètres, tels que décrits au Tableau 4.1. La Figure 4.1 illustre la construction globale de cette table de hachage.

Les Sections 4.1.1 à 4.1.5 décrivent comment la table de hachage « *tableRef »* est bâtie, en fonction des classes grammaticales.

Tableau 4.1 : Contenu du vecteur « *param »* selon la classe grammaticale

Classe grammaticale	Paramètre 0[5]	Paramètre 1	Paramètre 2	Paramètre 3
Verbe	temps	personne		
Adjectif	genre	nombre		
Nom commun	genre	nombre		
Adverbe				
Déterminant	genre	nombre	personne	
Pronom	genre	nombre	personne	cas
Préposition				
Conjonction				
Interjection				

[4] Le nom de variable « POS » a été choisi pour représenter le terme « *part of speech* » en anglais, qu'on peut associer à la classe grammaticale

[5] Avec Java, les indices débutent à zéro

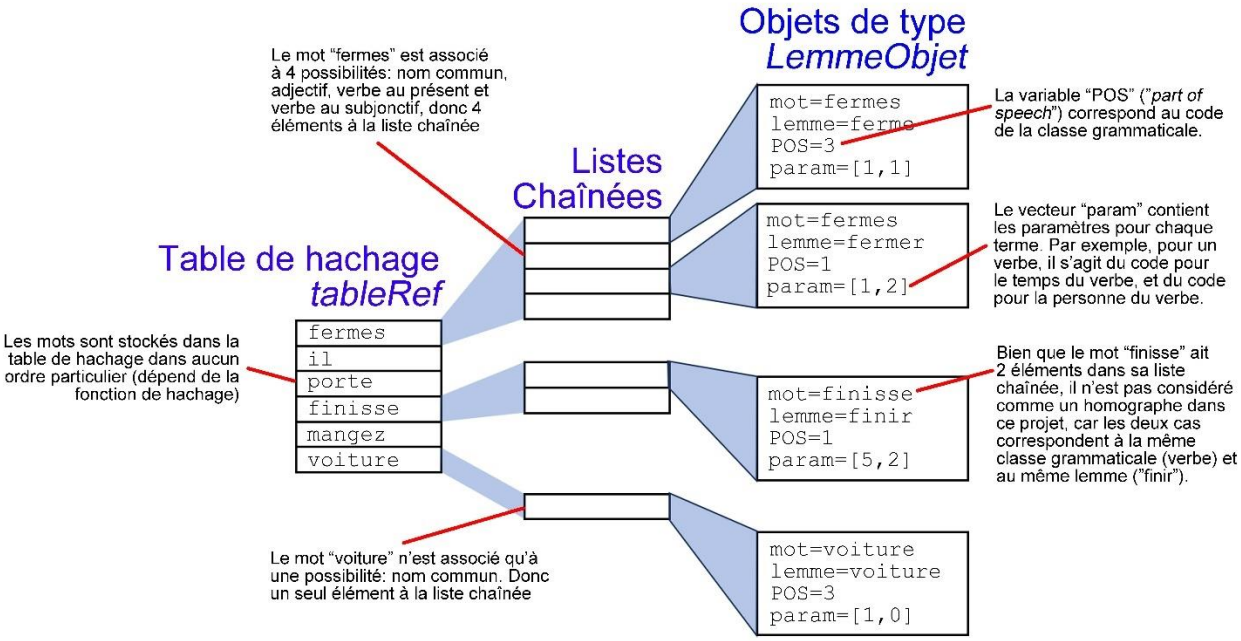


Figure 4.1 : Illustration de la construction de la table de hachage « *tableRef* »

4.1.1. Tableaux pour verbes conjugués

Le Bescherelle (2012) fournit un tableau comprenant près de 7000 verbes classés en ordre alphabétique, soit une liste quasi exhaustive des verbes de la langue française. Tel que mentionné à la Section 3.1.3.1, à chaque verbe est associé son modèle de conjugaison, ainsi que certains autres paramètres. En plus de ceux inclus au Bescherelle, d'autres paramètres ont été ajoutés pour le besoin de ce projet. Un fichier texte (délimité par tabulations) a donc été généré pour emmagasiner cette information. Un extrait de ce fichier est montré au Tableau 4.2.

Tableau 4.2 : Extrait du fichier texte contenant la liste des verbes, comprenant le modèle de conjugaison du Bescherelle, des valeurs de 0 (oui) ou 1 (non) pour les paramètres T (transitif), I (intransitif), P (pronominal), D (défectif), Ti (transitif indirect), AVOIR (temps composés avec l'auxiliaire « avoir »), ETRE (temps composés avec l'auxiliaire « être »), IL (verbe impersonnel), ATTRIBUT (verbe attributif), MODAL (verbe modal), puis PREP1, PREP2 et PREP3, les prépositions associées au verbe quand Ti=1

INFINITIF	MODELE	T	I	P	D	Ti	AVOIR	ETRE	Il	ATTRIBUT	MODAL	PREP1	PREP2	PREP3
aider	6	1	0	1	0	1	1	0	0	0	0	à	à+inf	
aigrir	19	1	1	1	0	0	1	0	0	0	0			
aiguiller	6	1	0	0	0	0	1	0	0	0	0			
aiguilleter	11	1	0	0	0	0	1	0	0	0	0			
aiguillonner	6	1	0	0	0	0	1	0	0	0	0			
aiguiser	6	1	0	1	0	0	1	0	0	0	0			
ailler	6	1	0	0	0	0	1	0	0	0	0			
aimanter	6	1	0	0	0	0	1	0	0	0	0			
aimer	6	1	0	1	0	0	1	0	0	0	1			

Une table de hachage, distincte de celle discutée à la Section 4.1, a été créée pour emmagasiner tous les verbes et leurs paramètres listés au Tableau 4.2. Tel que décrit plus haut, une table de hachage (appelée « *InfinitifH* [6]») permet un accès rapide aux données recherchées, grâce au concept de « clé ». Ici, la clé de la table « *InfinitifH* » est l'infinitif du verbe. Comme les tables de hachage ne permettent d'associer qu'une seule « valeur » pour chaque clé alors que plusieurs paramètres doivent être notés pour chaque verbe comme on le voit au Tableau 4.2, un objet Java a dû être créé. Cet objet (classe *VerbeInf*) regroupe tous les paramètres nécessaires. L'extrait de code 4.2 correspond à la classe *VerbeInf*. On note que les entiers 0 et 1 apparaissant au Tableau 4.3 sont emmagasinés comme valeurs booléennes dans la classe *VerbeInf*.

Extrait de code 4.2 : La classe « *VerbeInf* »

```
public class VerbeInf{

private String infinitif;
private int modele;
private boolean [] donnees;
private String [] prep;

// ---------- METHODE SET -------------------

public void Set (String infinitifSet, int modeleSet, boolean[] donneesSet, String[]
prepSet){

infinitif=infinitifSet;
modele=modeleSet;
donnees=donneesSet;
prep=prepSet;
}

// ---------- METHODES GET -------------------

public String GetInfinitif(){
return infinitif;
}
public int GetModele(){
return modele;
}
public boolean[] GetDonnees(){
return donnees;
}
public String[] GetPrep(){
return prep;
}
}
```

[6] Dans le nom de variable « InfinitifH », le « H » signifie « hachage »

En deuxième lieu, il a fallu créer les tableaux de conjugaison correspondant aux 82 modèles de conjugaison du Bescherelle (2012). Pour ce faire, un autre fichier texte a été créé, comprenant une colonne par modèle de verbe. Certains modèles ne réfèrent qu'à un seul verbe (exemple : « haïr »), tandis que d'autres modèles sont associés à de grandes quantités de verbes. Pour chaque modèle, on fournit la terminaison associée aux infinitifs pour ce modèle, ainsi que le nombre de lettres de cette terminaison. Parfois, le caractère « * » est utilisé lorsque plusieurs possibilités de lettres sont permises dans les terminaisons d'infinitif. Par la suite, chaque ligne du tableau correspond à un temps et une personne de verbe. Le Tableau 4.3 donne l'exemple du contenu de ce fichier pour quelques modèles. Pour les Modèles 1, 20 et 45 (« avoir », « haïr » et « pleuvoir »), le modèle ne correspond qu'à un seul verbe. Pour le Modèle 9, la présence du caractère « * » dans la terminaison signifie que le modèle en question s'applique à des verbes ayant différentes lettres à la position correspondant au « * ». Par exemple, pour le verbe « peser », le caractère « * » est associé à la lettre « s ». Tandis que pour le verbe « amener » qui se conjugue selon le même modèle, le caractère « * » correspond à la lettre « n ». Finalement, le Modèle 45 (verbe « pleuvoir ») illustre comment les formes non permises sont notées. Comme ce verbe ne se conjugue qu'à la troisième personne, le caractère « ! » a été inséré pour toutes les autres formes conjuguées, puisque celles-ci ne sont pas permises. L'algorithme n'insère donc pas d'élément à la table de hachage « *tableRef »* pour ces formes non permises.

L'algorithme passe ainsi à travers les milliers de verbes du Bescherelle, un à un, retrouvant le modèle de conjugaison de chacun. Ensuite, pour chacun de ces verbes, l'algorithme génère toutes ses formes fléchies permises, selon les modèles de conjugaison du Tableau 4.3. Une à une, chaque forme conjuguée est incorporée à la table de hachage « *tableRef »*, grâce à un énoncé Java « *put* ». À chaque verbe est associé l'entier « 1 » pour représenter cette classe grammaticale, dans la variable *POS* de l'objet « *LemmeObjet* » entré en valeur dans la table de hachage « *tableRef »*.

Dans bien des cas, une variation fléchie d'un verbe se retrouve à plus d'un temps et/ou plus d'une personne. Par exemple, comme on peut le voir au Tableau 4.3, on retrouve la forme fléchie « aime » du verbe « aimer » à la première personne et à la troisième personne du singulier de l'indicatif présent et du subjonctif présent. On retrouve aussi cette forme à la deuxième personne du singulier de l'impératif. La forme « aime » est donc associée à cinq possibilités de conjugaison. La valeur de la table de hachage « *tableRef* » pour la clé « aime » consiste donc en une liste chainée comprenant cinq éléments. Et tel que mentionné plus haut, le contenu de chaque élément de la chaîne est lui-même un objet (de type « *LemmeObjet »*) comportant l'information nécessaire pour définir chaque variante du mot fléchi. En tout, près de 350 000 formes de verbes conjugués sont créées et incluses dans la table de hachage « *tableRef* ».

Tableau 4.3 : Extrait du fichier texte contenant les formes de conjugaison du Bescherelle, pour quelques modèles. s=singulier, p=pluriel, m=masculin, f=féminin

# du modèle		1	6	9	20	45
Verbe modèle		avoir	aimer	peser	haïr	pleuvoir
Terminaison		!	er	e*er	haïr	pleuvoir
# lettres terminaison		5	2	4	4	8
Temps	Personne	Formes conjuguées				
Présent	1s	ai	e	è*e	hais	!
	2s	as	es	è*es	hais	!
	3s	a	e	è*e	hait	pleut
	1p	avons	ons	e*ons	haïssons	!
	2p	avez	ez	e*ez	haïssez	!
	3p	ont	ent	è*ent	haïssent	pleuvent
Imparfait	1s	avais	ais	e*ais	haïssais	!
	2s	avais	ais	e*ais	haïssais	!
	3s	avait	ait	e*ait	haïssait	pleuvait
	1p	avions	ions	e*ions	haïssions	!
	2p	aviez	iez	e*iez	haïssiez	!
	3p	avaient	aient	e*aient	haïssaient	pleuvaient
Passé simple	1s	eus	ai	e*ai	haïs	!
	2s	eus	as	e*as	haïs	!
	3s	eut	a	e*a	haït	plut
	1p	eûmes	âmes	e*âmes	haïmes	!
	2p	eûtes	âtes	e*âtes	haïtes	!
	3p	eurent	èrent	e*èrent	haïrent	!
Futur	1s	aurai	erai	è*erai	haïrai	!
	2s	auras	eras	è*eras	haïras	!
	3s	aura	era	è*era	haïra	pleuvra
	1p	aurons	erons	è*erons	haïrons	!
	2p	aurez	erez	è*erez	haïrez	!
	3p	auront	eront	è*eront	haïront	pleuvront
Subjonctif présent	1s	aie	e	è*e	haïsse	!
	2s	aies	es	è*es	haïsses	!
	3s	ait	e	è*e	haïsse	pleuve
	1p	ayons	ions	e*ions	haïssions	!
	2p	ayez	iez	e*iez	haïssiez	!
	3p	aient	ent	è*ent	haïssent	pleuvent
Subjonctif imparfait	1s	eusse	asse	e*asse	haïsse	!
	2s	eusses	asses	e*asses	haïsses	!
	3s	eût	ât	e*ât	haït	plût
	1p	eussions	assions	e*assions	haïssions	!
	2p	eussiez	assiez	e*assiez	haïssiez	!
	3p	eussent	assent	e*assent	haïssent	!
Impératif	2s	aie	e	è*e	hais	!
	1p	ayons	ons	e*ons	haïssons	!
	2p	ayez	ez	e*ez	haïssez	!
Conditionnel	1s	aurais	erais	è*erais	haïrais	!
	2s	aurais	erais	è*erais	haïrais	!
	3s	aurait	erait	è*erait	haïrait	pleuvrait
	1p	aurions	erions	è*erions	haïrions	!
	2p	auriez	eriez	è*eriez	haïriez	!
	3p	auraient	eraient	è*eraient	haïraient	pleuvraient
Participe présent		ayant	ant	e*ant	haïssant	pleuvant
Participe passé	ms	eu	é	e*é	haï	plu
	mp	eus	és	e*és	haïs	!
	fs	eue	ée	e*ée	haïe	!
	fp	eues	ées	e*ées	haïes	!

4.1.2. Tableaux pour noms communs

Les noms communs sont incorporés à la même table de hachage « *tableRef* » que les verbes. Chaque forme fléchie des noms communs sert donc de clé pour la table de hachage. Là encore, la valeur associée est une liste chainée, comprenant au moins un objet de la classe « *LemmeObjet* ». À chaque nom commun est associé l'entier « 3 » pour la variable *POS* de l'objet « *LemmeObjet »* pour représenter cette classe grammaticale.

Deux fichiers texte ont été créés pour répertorier tous les noms communs. Tout d'abord, un fichier comprenant les noms communs dits « animés ». Les noms animés existent aux deux genres ainsi qu'au pluriel et au singulier. On a par exemple « un policier », « des policiers », « une policière », et « des policières ». L'autre fichier texte comprend les mots dits « non-animés », n'existant qu'à un seul genre. On a par exemple « une chaise » et « des chaises », mais on ne peut pas dire « un chaise ». Tout comme pour les verbes, le caractère « ! » a été utilisé pour les assez rares cas où une forme du nom commun n'existe pas. Par exemple, il n'existe pas de singulier pour le mot « funérailles ». L'algorithme, lorsqu'il croise le caractère « ! », n'insère donc pas d'élément dans la table de hachage. Qu'ils soient animés ou non animés, les noms communs sont incorporés de la même façon que les verbes à la table de hachage « *tableRef »*. Seule la façon de lire les fichiers est différente. Un extrait du fichier texte de noms animés est fourni au Tableau 4.4. Les mots y sont inscrits dans l'ordre suivant : masculin singulier, masculin pluriel, féminin singulier, et féminin pluriel. On constate au Tableau 4.4 que dans certains cas, la forme féminine est identique à la forme masculine (comme pour le mot « acolyte »). Ces mots sont dits « épicènes ». Dans certains cas, les formes masculines et féminines sont très différentes (« beau » vs. « belle »). Un extrait du fichier de noms non-animés est fourni au Tableau 4.5. Dans ce cas, le nom au singulier est d'abord fourni, suivi du pluriel. Un code est ensuite inclus pour le genre du mot, où « 0 » signifie masculin et « 1 » signifie féminin. On constate que certains mots sont identiques au singulier et au pluriel (« accès ») et tel que mentionné plus haut, certains mots ne se retrouvent qu'à un seul nombre. Près de 8500 versions fléchies de noms communs non-animés et plus de 2000 versions fléchies de noms communs animés ont été générées pour ce projet. Ce nombre est de loin inférieur au nombre total de noms communs du dictionnaire, mais inclut néanmoins tous les noms communs apparaissant dans les corpus de référence sélectionnés pour ce projet.

Tableau 4.4 : Extrait du tableau de noms communs animés

Masculin singulier	Masculin pluriel	Féminin singulier	Féminin pluriel
abbé	abbés	abbesse	abbesses
académicien	académiciens	académicienne	académiciennes
accompagnateur	accompagnateurs	accompagnatrice	accompagnatrices
acolyte	acolytes	acolyte	acolytes
acteur	acteurs	actrice	actrices
adepte	adeptes	adepte	adeptes
beau	beaux	belle	belles

Tableau 4.5 : Extrait du tableau de noms communs non-animés

Singulier	Pluriel	Genre (0=masculin, 1=féminin)
abandon	abandons	0
abattement	abattements	0
abbaye	abbayes	1
abdomen	abdomens	0
accès	accès	0
!	mœurs	1

4.1.3. Tableaux pour adjectifs

Les adjectifs sont incorporés à la table de hachage « *tableRef* » de la même façon que les noms communs animés, puisque comme ceux-ci, les adjectifs se retrouvent généralement aux deux genres et aux deux nombres. À chaque adjectif est associé l'entier « 2 » à la variable *POS* de l'objet « *LemmeObjet* », pour représenter cette classe grammaticale. Un extrait du fichier texte d'adjectifs est fourni au Tableau 4.6. Les mots y sont inscrits dans l'ordre suivant : masculin singulier, masculin pluriel, féminin singulier, et féminin pluriel, tout comme pour les noms animés. Certains adjectifs sont identiques au masculin et au féminin (« abominable »), et certains autres sont identiques au singulier et au pluriel (« anglais »). Plus de 8000 versions fléchies d'adjectifs ont été générées pour ce projet. Une fois de plus, ce nombre d'adjectifs est de loin inférieur à toutes les formes existant au dictionnaire. Mais tous les adjectifs des corpus de référence se retrouvent dans le fichier.

Tableau 4.6 : Extraits du tableau d'adjectifs

Masculin singulier	Masculin pluriel	Féminin singulier	Féminin pluriel
abdominal	abdominaux	abdominale	abdominales
abject	abjects	abjecte	abjectes
abominable	abominables	abominable	abominables
abrupt	abrupts	abrupte	abruptes
abrutissant	abrutissants	abrutissante	abrutissantes
absent	absents	absente	absentes
absolu	absolus	absolue	absolues
anglais	anglais	anglaise	anglaises

4.1.4. Participes passés employés seuls

Les participes passés ont déjà été inclus à la table de hachage « *tableRef* », tel qu'on l'a décrit à la Section 4.1.1. Mais il est à noter qu'on peut retrouver des participes passés dans deux contextes principaux :

- Employés avec auxiliaire : « j'ai *fermé* la porte », « elle est *allée* à l'épicerie »
- Employés seuls : « la porte *fermée* », « du riz *frit* »

Lorsqu'ils sont employés seuls, donc non précédés de l'auxiliaire « être » ou de l'auxiliaire « avoir », les participes passés se comportent comme des adjectifs. Ils sont donc typiquement placés après le nom auquel ils se rapportent. Et tout comme les adjectifs, ils s'accordent en genre et en nombre avec ce nom. Pour ce projet, tous les participes passés, en plus d'être incorporés à la table de hachage « *tableRef* » comme formes verbales, y ont donc été aussi incorporés comme adjectifs. Ce « dédoublement » a été effectué pour deux raisons principales.

D'une part, puisque les adjectifs et les participes passés employés seuls occupent une fonction similaire dans la phrase, il est sensé et même souhaitable qu'ils soient considérés dans la même classe grammaticale du point de vue de l'algorithme de désambiguïsation. On retrouve en effet les adjectifs et les participes passés employés seuls typiquement aux mêmes endroits dans la phrase, donc entourant le même type de mots, ce qui fournit des indices sur leur classe grammaticale en présence d'homographes. On abordera la désambiguïsation plus en détail à partir de la Section 4.6.

D'autre part, considérer les adjectifs et les participes passés employés seuls comme faisant partie de la même classe grammaticale permet du coup d'employer ces deux types de mots lors de la génération de phrases aléatoires automatiquement lemmatisées à l'Étape 2 de ce travail. Ainsi, suite au mot « pomme » par exemple, l'algorithme de génération de phrases pourra opter aussi bien pour un adjectif (« rouge ») que pour un participe passé (« pourrie »).

Il découle de ce dédoublement que tous les participes passés seront considérés ici comme des homographes, puisqu'ils jouent soit le rôle d'une forme verbale lorsqu'employés avec un auxiliaire, soit le rôle d'un adjectif, lorsqu'employés seuls.

4.1.5. Tableaux pour autres mots

Les mots appartenant aux autres classes grammaticales (déterminants, adverbes, pronoms, prépositions, conjonctions et interjections) ont tous été regroupés dans un seul fichier texte. Ces mots sont aussi tous incorporés à la table de hachage « *tableRef* ». Comme pour les autres classes, à chaque mot sont associés son lemme, sa classe grammaticale (entier selon le Tableau 3.1), et ses paramètres. La liste de paramètres associés à chaque classe est fournie au Tableau 4.1. On retrouve près de 850 « autres mots » dans les banques de données lexicales. Le Tableau 4.7 fournit quelques exemples d'éléments de la banque de mots pour ces « autres mots ».

Tableau 4.7 : Extrait de la banque de données pour « autres mots ». Les codes de classes correspondent à ceux listés au Tableau 3.2. Pour le genre, 0=masculin, 1=féminin. Pour le nombre, 0=singulier, 1=pluriel, pour la personne, on a les nombres de 1 à 6 (1 à 3 au singulier, 4 à 6 au pluriel), et pour le cas, 0=nominatif, 1=accusatif et 2=datif. La valeur « -1 » signifie que le paramètre n'est pas applicable

Mot	Lemme	Classe	Genre	Nombre	Personne	Cas
apparemment	apparemment	4	-1	-1	-1	-1
à	à	7	-1	-1	-1	-1
car	car	8	-1	-1	-1	-1
adieu	adieu	9	-1	-1	-1	-1
la	le	51	1	0	-1	-1
les	le	51	0	1	-1	-1
les	le	51	1	1	-1	-1
mes	mon	53	0	1	1	-1
notre	mon	53	0	0	1	-1
sa	son	53	1	0	3	-1
je	je	61	0	0	1	0
les	les	61	0	1	6	1
lui	lui	61	1	0	3	2
me	me	61	0	0	1	1
moi	moi	61	0	0	1	2
soi	soi	61	0	0	3	2
mienne	mien	63	1	0	1	-1

Il est à noter qu'une fois que tous les mots de toutes les classes grammaticales ont été incorporés à la table de hachage « *tableRef* », certains d'entre eux sont associés à des listes chainées comprenant plus d'un élément. Ces mots ne sont pas nécessairement considérés comme des homographes. Si on se réfère à la définition d'homographe adoptée pour ce projet (voir Section 3.2.2), pour être considéré comme un homographe pour ce projet, un mot doit être possible dans plus d'une classe grammaticale ou encore être associé à plus d'un lemme. Le mot « aime » par exemple, bien que sa liste chainée comporte cinq éléments, n'est pas considéré comme un homographe pour ce projet, puisque les cinq variations de ce mot sont toutes des verbes, et font aussi toutes référence au même lemme « aimer ».

4.2. Préparation du texte du corpus de référence

La lemmatisation du corpus de référence, servant à en extraire la banque de lemmes à utiliser pour générer des textes aléatoires, se fait dans le cadre de ce projet avec un outil informatique. Il est donc essentiel que le corpus de référence soit disponible dans un format qui soit accessible et lisible par un outil informatique. Les prochaines sections décrivent comment le texte du corpus de référence est chargé et traité pour le rendre accessible aux algorithmes de lemmatisation et de génération de phrases aléatoires.

4.2.1. Traitement manuel du corpus de référence

Les textes de référence identifiés à la Section 3.4 ont dû être traités pour rendre leur format compatible à l'analyse par le lemmatiseur développé pour ce projet. Ce traitement a d'abord consisté à transférer les documents sous un format « texte » (code ASCII), facilement interprétable dans un langage de programmation. Comme les documents originaux étaient en format Microsoft Word, il n'a suffi que d'enregistrer les fichiers en format texte.

Un traitement manuel mineur a par la suite été effectué. En effet, dans le cas du roman « Le Rouge et le Noir », chaque chapitre débute par une citation, souvent en langue étrangère (italien ou anglais). Ces citations en langues étrangères ont tout simplement été éliminées pour ne pas nuire à la lemmatisation. De plus, chaque chapitre du roman de science-fiction aussi utilisé comme corpus de référence commence par le mot « chapitre », suivi du numéro du chapitre. Ces entêtes de chapitre ont été retirées manuellement. Ne pas avoir retiré ces entêtes aurait eu pour conséquence d'artificiellement augmenter la fréquence d'apparition du mot « chapitre », sans qu'il fasse réellement partie du texte. Comme autre conséquence, l'algorithme de désambiguïsation des homographes aurait ainsi identifié un grand nombre de « phrases » de deux mots dont le nombre correspond au nombre total de chapitres, débutant par un nom commun, ce qui aurait biaisé l'analyse. Bien que ces conséquences ne soient pas si fâcheuses, ces entêtes ont tout de même été retirées.

D'autres modifications manuelles peuvent être nécessaires pour d'autres textes. Par exemple, si les textes sont d'abord extraits de fichiers « PDF », certains caractères spéciaux peuvent s'introduire dans le fichier texte, ce qui peut, là encore, fausser l'analyse de différentes façons.

4.2.2. Charger le texte du corpus de base

Une fois le fichier texte du corpus de référence créé tel que décrit à la Section 4.2.1, il faut le charger en mémoire pour en faire l'analyse. Pour ce faire, le fichier externe, en format dit ASCII, doit être lu par l'algorithme, puis être transféré dans une variable ou objet permettant son traitement dans le langage Java. Une façon courante d'enregistrer en mémoire le contenu d'un fichier texte est d'avoir recours à un objet du type « *String* ». Cependant, les variables-objet du type String en Java sont dites « *immutables* ». Autrement dit, elles ne peuvent être modifiées autrement qu'en créant un nouvel objet à chaque modification. Bien que cette opération soit invisible au programmeur, elle n'est pas sans conséquence, à la fois pour la gestion de la mémoire et pour la rapidité d'exécution d'un programme. Supposons par exemple qu'un fichier texte ASCII comporte les caractères suivants : « Bonjour la Terre ». Si on lit ce fichier un caractère à la fois, et qu'on ajoute chaque caractère un à un pour former une chaîne de caractères de type *String* comportant toutes ces lettres, un grand nombre d'objets *String* se retrouveront à être créés en mémoire. La Figure 4.2 illustre ce concept.

Si on ne tient pas compte ici des espaces, par simplicité, on constate à la Figure 4.2 qu'un total de 14 objets a été créé (un objet pour chaque ligne), correspondant au nombre de lettres dans le fichier lu. Mais plus important que le nombre d'objets est la taille totale en mémoire, proportionnelle au nombre total de lettres pour tous ces objets. Pour le segment « Bonjour la Terre», ce nombre total est de 1+2+3+4+5+6+7+… +14 = 105 lettres. En effet, le tableau de la Figure 4.2 contient 105 cases. On peut facilement s'imaginer l'espace mémoire requis pour un texte de la taille du corpus de référence traité dans ce projet, comprenant près de 200 000 mots. En utilisant une moyenne de 10 lettres par mots pour la langue française, on estime donc qu'environ deux millions de caractères sont compris dans le corpus de référence. Si on lit ce corpus lettre par lettre dans un objet *String* comme on l'a fait plus haut pour le segment de phrase « Bonjour la Terre », on se retrouve avec deux millions d'objets, qui comprendront environ $2x10^{12}$ caractères au total, en utilisant l'équation fournie à la Figure 4.2. Cette création d'autant d'objets, dont certains de grande taille, ralentit énormément l'exécution. La Figure 4.3 illustre graphiquement comment le nombre de caractères en mémoire évolue en fonction du nombre de caractères dans le corpus, lorsqu'on charge le fichier avec une variable de type « *String* ».

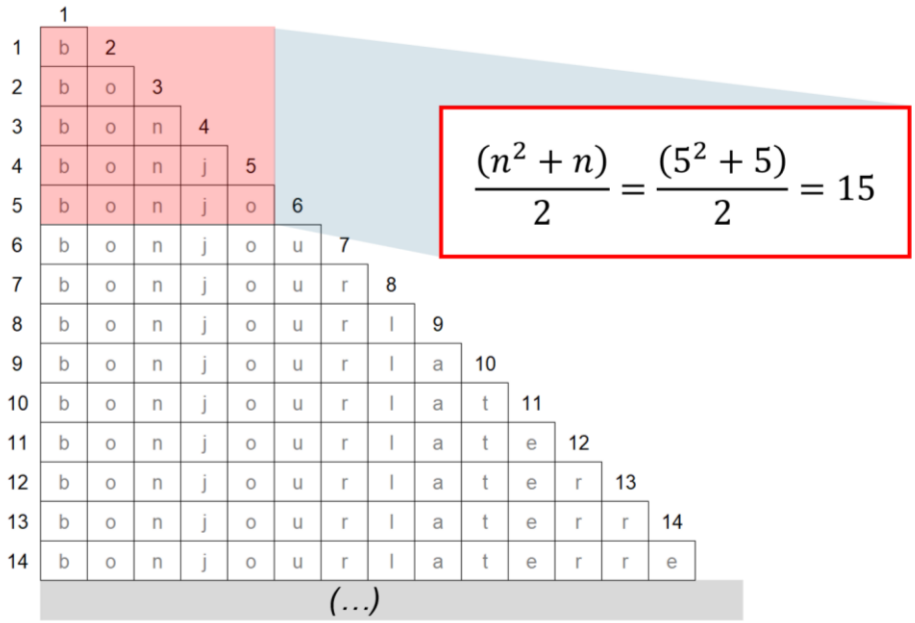


Figure 4.2 : Illustration du transfert d'un fichier texte à une variable de type « *String* », mettant en évidence une surutilisation d'espace mémoire pour y arriver

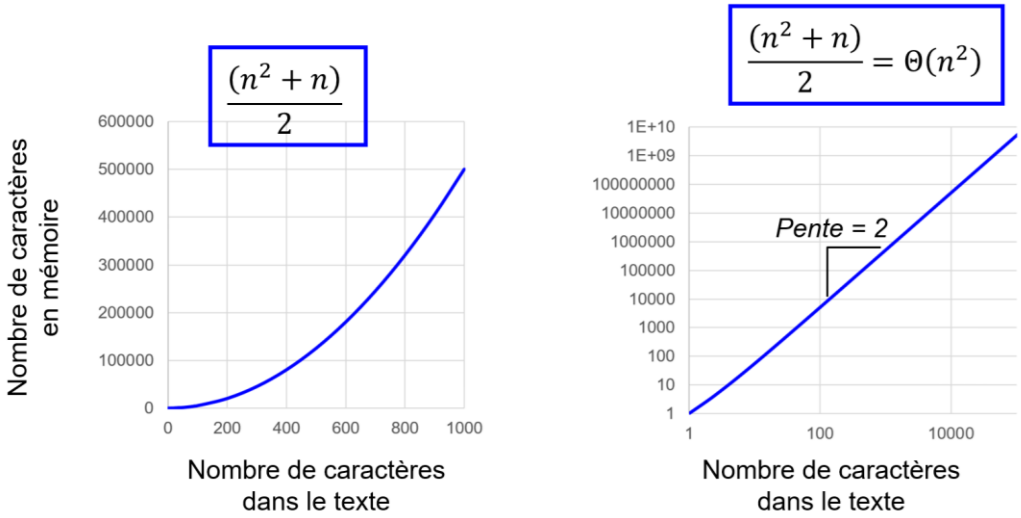


Figure 4.3 : Graphique du nombre de caractères en mémoire lors du chargement de fichiers texte avec une variable de type « String » en fonction du nombre de caractères dans le texte. À gauche, une échelle standard, à droite, une échelle logarithmique mettant en évidence l'ordre algorithmique de 2

Dans le cas qui nous intéresse, la rapidité d'exécution est critique, puisque nous travaillons avec des banques de données contenant des centaines de milliers de mots. On a donc eu recours à une structure de données autre que « *String* », soit la structure « *StringBuilder* » de Java. Le grand avantage de cette structure est qu'il n'est pas nécessaire de créer un nouvel objet de type *StringBuilder* quand l'objet est modifié, puisque l'objet *String Builder* n'est pas immutable. En bout de ligne, au lieu de créer deux millions d'objets comportant en tout $2x10^{12}$ caractères, on se retrouve plutôt à ne créer qu'*un seul objet* contenant deux millions de caractères. Cela représente un gain énorme en espace mémoire et en rapidité d'exécution. De façon générale, les algorithmes bâtis pour ce projet utilisent le type d'objet *StringBuilder* plutôt que *String*, dès que des modifications multiples sont nécessaires sur une chaîne de caractères.

Le chargement complet du corpus est donc effectué avec des opérations de lecture de fichiers de base de Java, un caractère à la fois, transféré dans un objet de type *StringBuilder*. Une fois cet objet créé, on peut, au besoin, le transformer en un objet de type *String* en une seule opération ne créant qu'un seul nouvel objet. Le texte complet du corpus de référence se retrouve donc emmagasiné dans un seul objet, disponible pour les opérations ultérieures de manipulation de chaînes de caractères, telles que la recherche, le remplacement et l'effacement.

4.2.3. Nettoyer le texte du corpus de base

Une fois l'étape du chargement du corpus de référence franchie, le fichier texte est davantage simplifié, en y enlevant les sauts de paragraphes et la plupart des signes de ponctuation, tels que les points-virgules, guillemets, apostrophes, afin que l'algorithme se concentre sur les mots individuels. Mais la ponctuation est tout de même utilisée en parallèle pour déterminer le début et la fin des phrases, pour repérer les cooccurrences dans chaque phrase et pour faciliter la désambiguïsation des homographes. On conserve donc les points à la fin des phrases. Et comme les phrases peuvent aussi se terminer par des points d'exclamation ou des points d'interrogation, on remplace ceux-ci par des points. On note aussi la position des virgules car celles-ci seront utiles pour la désambiguïsation, puisque des mots appartenant à certaines classes grammaticales sont plus souvent observés directement devant ou directement suivant une virgule.

Les majuscules sont aussi remplacées par des minuscules puisque la table de hachage « *tableRef* » ne comporte que des minuscules. Il est toutefois à noter que les majuscules, lorsque présentes ailleurs qu'en tout début de phrase, servent à identifier la présence de noms propres dans le corpus de référence. L'identification des noms propres et l'algorithme associé sont discutés à la Section 4.2.4.

Un vecteur est donc créé comportant tous les caractères de ponctuation. La méthode « *replace* » de Java est appliquée à la variable objet *String* du texte, pour remplacer tous ces caractères par des espaces, ou encore les points d'exclamation et d'interrogation par des points ordinaires, tel que discuté plus haut.

Tous les chiffres et tous les mots contenant des chiffres sont ensuite retirés. Ce nettoyage s'effectue avec l'aide d'expressions régulières, aussi appelées « *regex* », de l'anglais « *regular expressions* ». Celles-ci permettent d'identifier rapidement toutes les unités du texte incluant des chiffres. La méthode entière pour retirer les chiffres est incluse dans l'extrait de code 4.3, où les instructions directement reliées aux expressions régulières ont été surlignées en jaune.

Extrait de code 4.3 : La méthode pour retirer les chiffres. Les instructions directement reliées aux expressions régulières ont été surlignées en jaune

```
public static String EnleverChiffres(String corpusBrut){

HashMap<String,Integer> listeChiffres =new HashMap<>();
boolean existe;
Pattern pattern = Pattern.compile(" [0-9]+ ");
Matcher m = pattern.matcher(corpusBrut);
String chiffre;
int index, freq;

// Créer liste des chiffres sans répétitions
while (m.find()){
    index=corpusBrut.indexOf(" ",m.start()+1);
    chiffre=corpusBrut.substring(m.start()+1,index);
    existe=listeChiffres.containsKey(chiffre); // Est-ce key ou value?
    if(!existe){
        listeChiffres.put(chiffre, 1);}
    else{
        freq=listeChiffres.get(chiffre);
        listeChiffres.replace(chiffre, freq+1);}
}  // fin du while

// Maintenant identifiés, enlever les chiffres du texte
Set<String> keys = listeChiffres.keySet();
for(String key: keys){
    corpusBrut=corpusBrut.replace(" "+key+" "," "); // rajouté espace
}
return corpusBrut;
}
```

4.2.4. Extraction et traitement des noms propres

Le « Dico en ligne Le Robert » (2024) définit le nom propre comme désignant « un individu (ou un groupe d'individus), un lieu ou une chose unique, contrairement au nom commun qui désigne des classes de personnes, de lieux, de choses ou d'abstractions ». Il ne peut exister de banque de données exhaustive des noms propres, puisque de nouveaux noms propres sont créés continuellement, au gré de la créativité des humains. Les noms propres commencent tous par des majuscules, ce qui les rend facilement identifiables. Un algorithme d'identification des noms propres débute donc par la recherche de tous les mots d'un texte débutant par une majuscule.

Cependant, le premier mot de toute phrase débute aussi par une majuscule, peu importe sa classe grammaticale. Par exemple, dans la phrase « Jean mange du tofu », le mot « Jean » est un nom propre. Mais dans la phrase « Parfois Jean mange du tofu », le mot « Parfois » n'est pas un nom propre. La présence d'une majuscule au début d'un mot n'est donc pas un indice suffisant pour identifier un nom propre, quand le mot en question est le premier de la phrase. Le défi d'un algorithme d'identification des noms propres est donc d'identifier, parmi les premiers mots de toutes les phrases, lesquels sont des noms propres.

L'approche adoptée pour ce projet, consiste à d'abord rechercher dans le texte, tous les mots commençant par une majuscule, mais ne débutant pas une phrase (on réfère à cette étape comme le « Passage 1 »). Pour s'assurer qu'un mot n'est pas en début de phrase, on vérifie qu'il

n’est pas précédé d’un point, à l’aide d’une expression régulière. Une table de hachage comportant ces mots est créée, dont la clé est le nom propre lui-même. Dans le contexte présent, la valeur de la table est la fréquence d’apparition du nom propre dans le texte. L’attrait d’une table de hachage ici est qu’elle permet facilement d’éviter les doublons, grâce à la fonction « *exist* » appliquée aux clés de la table. De plus, la table de hachage permet ensuite de facilement comparer les mots du texte avec les clés.

À l’étape suivante (« Passage 2 »), une fois cette banque initiale de noms propres générée, on s’attaque aux premiers mots des phrases, dont on retire d’abord les majuscules. Si un tel mot fait partie de la table de hachage « *tableRef* » décrite à la Section 4.1 à une classe autre que nom propre, on en déduit que ce mot n’est *pas* un nom propre. Sinon, on considère ce mot comme étant un nom propre. Chaque nom propre, dès qu’identifié tel quel, est ajouté à la table de hachage « *tableRef* », avec le code « 32 » associé aux noms propres. La Figure 4.4 illustre avec un exemple fictif mais concret, le fonctionnement de l’algorithme d’identification des noms propres.

En parallèle à l’identification des noms propres, tel qu’on l’a mentionné plus haut, l’algorithme compile aussi leurs fréquences d’apparition dans le corpus. La liste complète des noms propres identifiés dans le texte est fournie en sortie. Cette liste permet de potentiellement y repérer des mots qui ne sont pas des noms propres, mais qui ont été identifiés tel quels, parce qu’ils n’avaient pas été inclus préalablement dans la table de hachage « *tableRef* ». On peut donc rajouter ces mots à « *tableRef* » en fonction de leur classe grammaticale, et effectuer l’analyse de nouveau.

L’algorithme d’identification des noms propres décrit ici n’est pas parfait. Si par exemple, le nom propre « Rose » est employé dans le texte, mais qu’il se retrouve uniquement en début de phrase, il sera alors associé uniquement au nom commun ou à l’adjectif « rose ». De plus, dans ces cas où un nom propre a la même graphie qu’un autre mot (homographe), le calcul des fréquences d’apparition est affecté. Mais cette faiblesse de l’algorithme est ici acceptée.

Lors du nettoyage du texte, les noms propres identifiés comme tels sont modifiés pour inclure un astérisque (« * ») devant eux. Ainsi, la présence de ces astérisques permet ensuite à l’algorithme de lemmatisation de facilement les identifier. Il est à noter que pour le projet actuel, les noms propres ne sont compilés qu’à titre de « référence ». Ceux-ci n’ont pas été utilisés pour générer des phrases aléatoires à l’Étape 2 du mémoire.

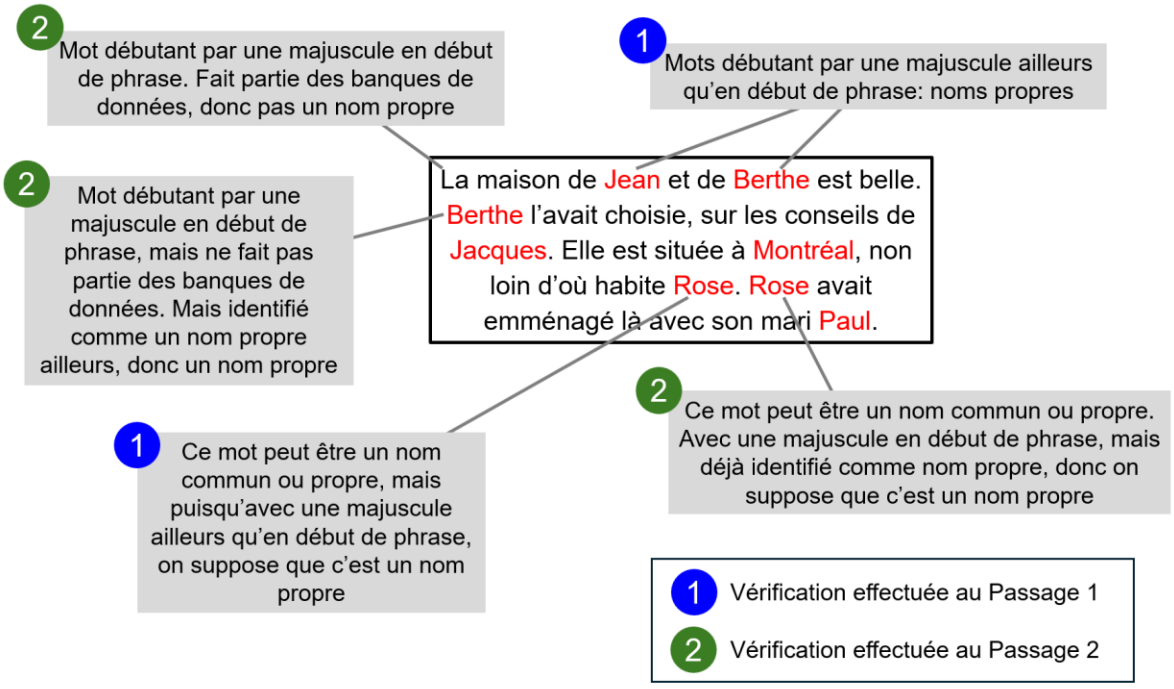


Figure 4.4 : Illustration de l'algorithme pour identifier les noms propres dans le corpus de référence

4.3. Algorithme pour la lemmatisation de base du corpus de référence

La lemmatisation de base s'effectue à partir du corpus de référence nettoyé, suivant les étapes décrites plus haut à la Section 4.2. Une boucle s'effectue pour analyser, de façon séquentielle, chaque mot du texte. On vérifie si chaque mot correspond à une des clés de la table de hachage « *tableRef* », ce qui équivaut à se demander si ce mot existe dans les banques de données lexicales créées pour le projet. C'est ici que l'avantage de l'utilisation d'une table de hachage pour « *tableRef* » est le plus grand, car rechercher un mot parmi une banque en contenant quelques centaines de milliers serait très long en parcourant par exemple tous les éléments d'un vecteur (« *array* ») standard de façon séquentielle.

Quand un mot du texte ne correspond à aucune clé de la table de hachage, autrement dit, si aucune clé ne lui correspond, et que ce mot n'est pas non plus un nom propre, on en déduit qu'il n'est pas classé. On lui attribue alors la classe grammaticale « 0 ». Comme tous les mots non classés sont répertoriés et fournis en sortie, il est ensuite possible de les analyser et de les ajouter au besoin dans les banques lexicales, en fonction de leur classe grammaticale, puis de réinitialiser l'analyse. À la Section 4.6.6, on décrit un algorithme « *guesser* [7]» servant à déterminer la classe grammaticale la plus probable d'un mot non classé, en utilisant l'algorithme de désambiguïsation.

Si au contraire, le mot du texte sous analyse se retrouve dans la table de hachage, autrement dit, si une clé lui correspond, on associe à ce mot la valeur de la table de hachage lui correspondant. À titre de rappel, la valeur de la table de hachage « *tableRef* » est un objet de type « *LemmeObjet* », contenant une liste chaînée de toutes les possibilités pour le mot en question. Par exemple, au mot « aime » est associée une liste chaînée de cinq éléments, tel que discuté à la Section 4.1.1, car il existe cinq formes de conjugaisons l'impliquant. De la même façon, au mot « pas » est associé une liste chaînée de deux éléments, puisque ce mot peut être ou bien un

[7] Le terme anglais « *guesser* » dans ce contexte est couramment utilisé par les linguistes francophones. Voir par exemple Vergne (2002)

adverbe, ou bien un nom commun. Finalement, pour le mot « lentement », la liste chaînée ne contient qu'un seul élément, puisque ce mot n'est pas un homographe. Les objets « *LemmeObjet* » contiennent l'information grammaticale correspondant à chaque possibilité (lemme, classe grammaticale et autres paramètres).

Aussi, comme les points n'ont pas été enlevés lors du nettoyage de la ponctuation, ils servent ici à identifier les fins de phrase. Ce repérage est critique pour les étapes d'identification des cooccurrences (Section 4.4) et pour la désambiguïsation des homographes (Section 4.6), puisque leurs algorithmes associés s'effectuent uniquement au niveau de la phrase. La boucle de lemmatisation génère donc un vecteur identifiant l'indice de tous les premiers mots de phrase, grâce à la position des points. La présence des points sert aussi à compter le nombre de phrases du corpus de référence, et à extraire le nombre de mots dans chacune. Un histogramme du nombre de mots par phrase est bâti en parallèle. Le nombre total de mots dans le corpus est aussi compilé dans cette boucle. Le Tableau 4.8 illustre toutes les informations fournies en sortie de la boucle de lemmatisation. Finalement, la Figure 4.5 résume le fonctionnement de l'algorithme de lemmatisation de base, qui exclut pour l'instant toute désambiguïsation des homographes.

Tableau 4.8 : Informations en sortie de la boucle de lemmatisation

Nom d'objet/variable et type	Description
motOrdreRaw (String [ ])	La liste de tous les mots tels qu'ils apparaissent dans le texte
motDetail (LemmeObjet [ ])	Valeur de l'objet « *LemmeObjet* » pour chaque mot du texte. Inclut les informations grammaticales de tous les homographes possibles
phrase (int [ ])	Numéro séquentiel de la phrase à laquelle appartient le mot
debutPhraseEnt (int [ ])	Position du début de chaque phrase
nPhrase (int)	Nombre de phrases dans le corpus
nMotsPhrase (int [ ])	Nombre de mots dans chaque phrase
histCorpus (int [ ])	Nombre de phrases comportant chaque nombre de mots.
nMots (int)	Le nombre de mots total dans le corpus

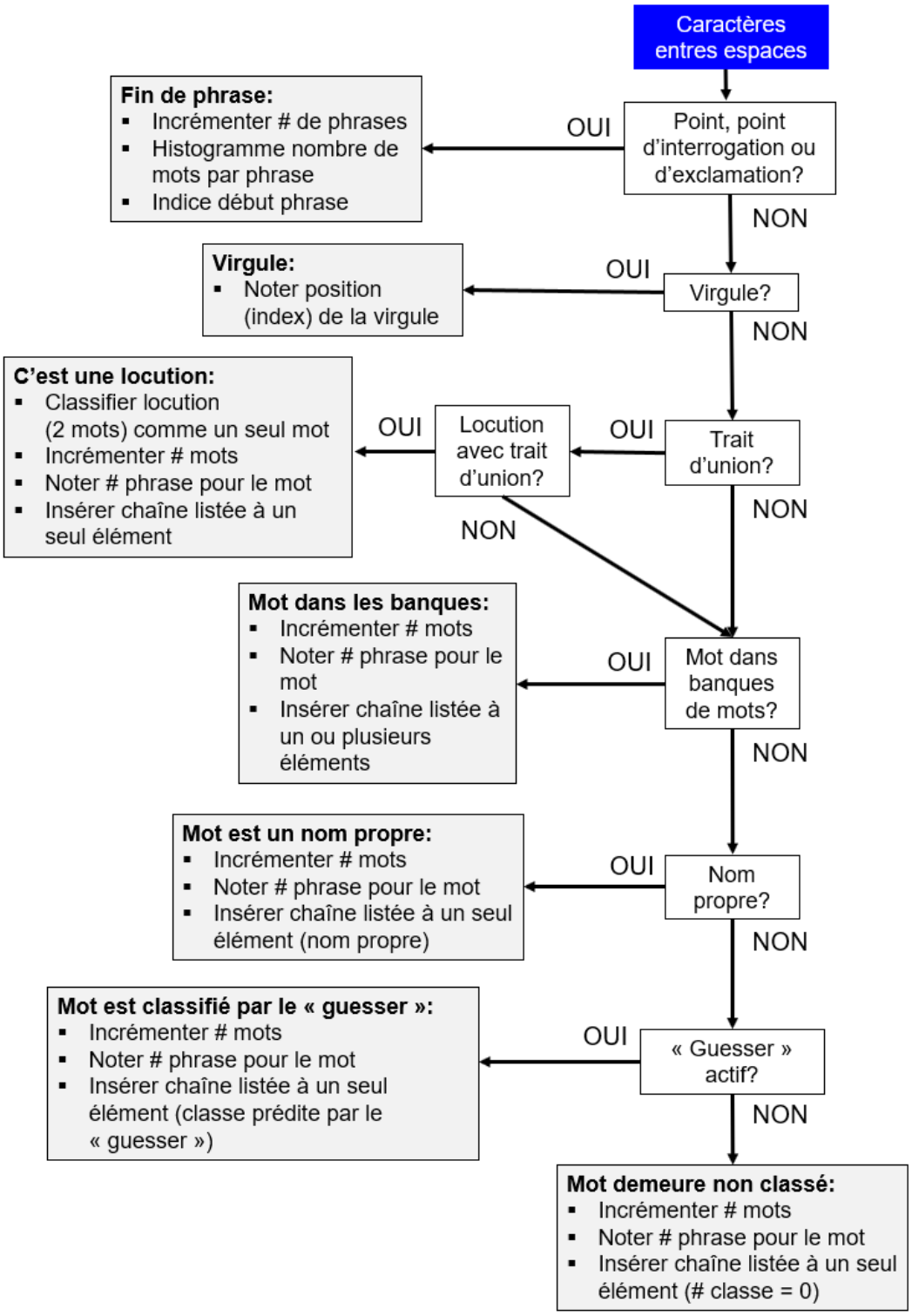


Figure 4.5 : Illustration de l'algorithme de lemmatisation de base (n'incluant pas encore la désambiguïsation des homographes)

4.3.1. Tableaux de fréquences

Une fois la boucle de lemmatisation telle que décrite à la Section 4.3 terminée, l'algorithme compile les fréquences d'apparition des mots, des temps de verbes et des personnes de verbes. Il faut rappeler que ces fréquences représentent un des éléments-clé en sortie de l'Étape 1. En effet, la sélection des mots, des temps et des personnes de verbes à l'Étape 2 (génération aléatoire de phrases) se fait en fonction des fréquences observées dans le corpus de référence. Ainsi, un mot apparaissant souvent dans le corpus de référence apparaîtra souvent aussi dans les phrases générées aléatoirement à l'Étape 2. De la même manière, si par exemple le temps de verbe le plus usité dans le corpus est l'indicatif présent, il en sera de même pour les phrases générées aléatoirement, à moins de modifier manuellement les paramètres.

Il est toutefois important de mentionner que jusqu'à présent, aucune désambiguïsation des homographes n'a encore eu lieu. C'est donc dire que les homographes, qui sont très nombreux, ne sont pas encore associés à une seule de leurs variations en particulier. Quand vient le temps de compiler les fréquences, deux approches sont donc adoptées pour tenir compte de cette ambiguïté.

La première approche consiste à considérer chacune des possibilités d'homographes. Ainsi, un mot associé à deux possibilités, comme le mot « pas », est considéré à la fois comme un adverbe et comme un nom commun. Autrement dit, sa présence fait augmenter à la fois la fréquence des adverbes, et la fréquence des noms. De la même façon, le mot « aime », fait augmenter la fréquence des temps « présent », « subjonctif présent » et « impératif », ainsi que les fréquences des trois personnes du singulier. La deuxième approche consiste à ne s'attarder qu'aux formes uniques (non ambiguës). Dans un tel cas, le mot « pas » n'est pas considéré, et le mot « aime » ne contribue pas aux calculs de fréquences des temps et des personnes de verbes.

Tableau 4.9 : Informations en sortie de l'algorithme de compilation des fréquences

Nom d'objet/variable et type	Description
freqAmbigu (table de hachage, clé=lemme, valeur=fréquence)	La fréquence d'apparition de chaque lemme (version non fléchie d'un mot), en tenant compte de toutes les possibilités d'homographes
freqNonAmbig (table de hachage, clé=lemme, valeur=fréquence)	La fréquence d'apparition de chaque lemme (version non fléchie d'un mot), en tenant compte uniquement des cas non ambigus (non homographes)
tempsArray (int [ ])	La fréquence d'apparition de tous les temps de verbe en tenant compte de toutes les possibilités. L'indice correspond au code du temps de verbe (Tableau 3.4)
tempsArrayNA (int [ ])	La fréquence d'apparition de tous les temps de verbe en tenant compte uniquement des cas non ambigus. L'indice correspond au code du temps de verbe (Tableau 3.4)
personneArray (int [ ])	La fréquence d'apparition de toutes les personnes de verbe en tenant compte de toutes les possibilités. L'indice correspond à la personne du verbe (Tableau 3.5)
personneArrayNA (int [ ])	La fréquence d'apparition de toutes les personnes de verbe en tenant compte uniquement des cas non ambigus. L'indice correspond à la personne du verbe (Tableau 3.5)

Pour le projet actuel, des statistiques sont compilées pour ces deux approches. On peut considérer que les véritables fréquences, celles qui seraient compilées pour le texte parfaitement désambiguïsé, se retrouvent à quelque part entre les extrêmes correspondant à ces deux approches. Cependant, une fois la désambiguïsation effectuée (voir la Section 4.6), les deux approches fournissent le même résultat, puisqu'il n'y a alors plus d'ambiguïté dans le texte. La méthode d'analyse de fréquence peut donc être effectuée de nouveau après la désambiguïsation, pour obtenir de bien meilleures estimations des fréquences. Le Tableau 4.9 illustre toutes les informations fournies en sortie l'algorithme de compilation des fréquences. La Figure 4.6 quant à elle, illustre le procédé de compilation des fréquences des temps de verbes, et la Figure 4.7 illustre le cas des personnes de verbes.

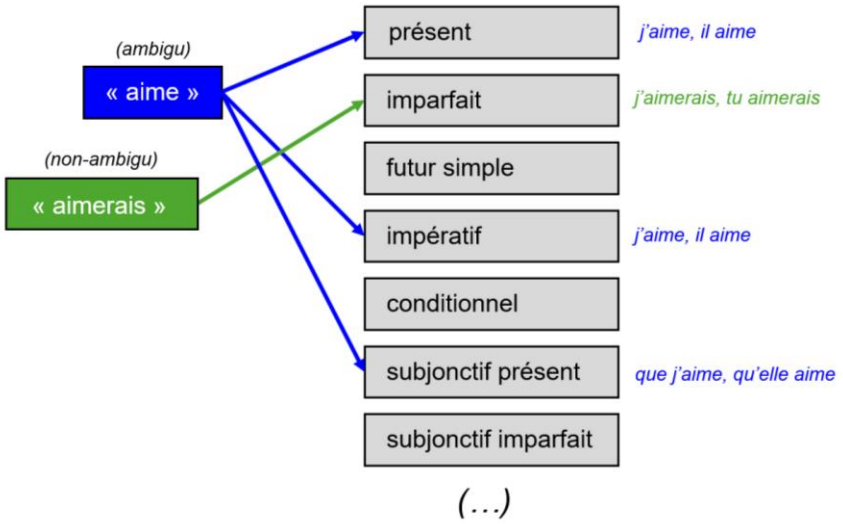


Figure 4.6 : Illustration du processus de compilation des fréquences des temps de verbes. On y distingue un exemple « ambigu » (« aime ») et un exemple « non-ambigu » (« aimerais »). Il est à noter que même si la forme verbale « aimerais » correspond à deux formes conjuguées, on considère cette forme comme « non ambiguë » du point de vue du temps de verbe, puisque les deux formes conjuguées correspondent à ce même temps de verbe

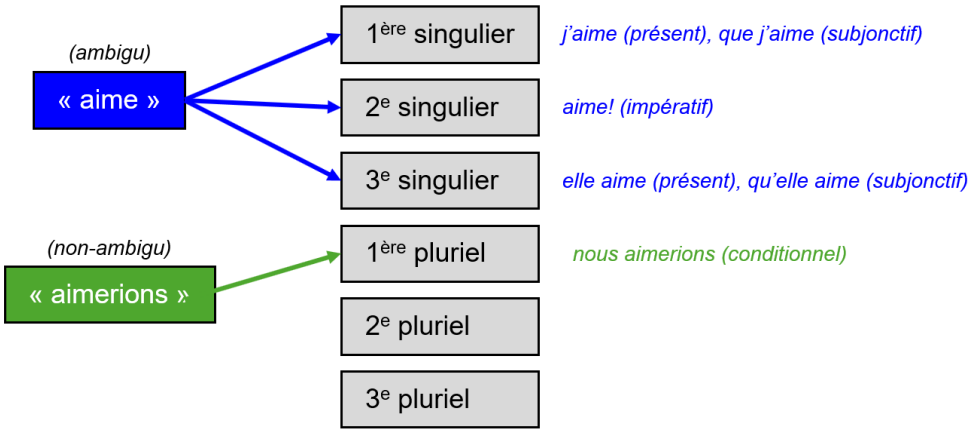


Figure 4.7 : Illustration du processus de compilation des fréquences des personnes de verbes. On y distingue un exemple « ambigu » (« aime ») et un exemple « non ambigu » (« aimerions »)

4.4. Génération des tables de cooccurrences

La pertinence des tables de cooccurrences et l'approche générale pour les bâtir ont été discutées à la Section 3.3. Nous allons maintenant décrire comment les tables de hachage ont été utilisées pour faciliter la construction de ces tables de cooccurrences, en réutilisant les mêmes phrases utilisées au Chapitre 3 :

Phrase 1 : Les voitures rouges roulent rapidement.

Phrase 2 : Ces voitures passent sur la lumière rouge.

Comme énoncé précédemment, l'analyse des cooccurrences ne porte pour ce projet que sur quatre classes grammaticales : les verbes, les noms, les adjectifs et les adverbes. Dans ces deux phrases, nous allons donc extraire uniquement ces mots, avec un code de couleurs :

Phrase 1 : Les voitures rouges roulent rapidement.
Phrase 2 : Ces voitures passent sur la lumière rouge.

Oubliant les autres mots, nous obtenons :

Phrase 1 : voitures rouges roulent rapidement
Phrase 2 : voitures passent lumière rouge

Ensuite, nous ne considérons que les lemmes de chacun de ces mots :

Phrase 1 : voiture rouge rouler rapidement
Phrase 2 : voiture passer lumière rouge

Pour l'analyse des cooccurrences, nous ne considérons dans ce projet que les mots se retrouvant dans une phrase donnée. Commençons donc avec la Phrase 1. Pour le lemme « voiture », on retrouve donc 3 cooccurrences : « rouge », « rouler » et « rapidement ». Celles-ci appartiennent à des classes grammaticales différentes. Commençons avec le lemme « voiture » pour bâtir les tables de hachage correspondantes.

Une nouvelle table de hachage (« *lemmeOccur* ») est donc créée. Celle-ci a donc comme première clé la chaîne de caractères « voiture ». La valeur correspondant à cette clé est un objet du type « O*ccurObjet* ». Cet objet appartient à une classe contenant quatre tables de hachage. Ces quatre tables de hachage sont appelées « *Noms* », « *Adjectifs* », « *Verbes* » et « *Adverbes* ». Leurs clés correspondent aux cooccurrences (lemmes, donc des « *String* »), et leurs valeurs correspondent à la fréquence à laquelle on retrouve la cooccurrence (valeurs entières). La classe « *OccurObjet* », en plus des méthodes de type « *Set* » et « *Get* », inclut la méthode « *AjouteMot* », qui, comme son nom l'indique, sert à ajouter un mot dans une des tables s'il n'y est pas déjà, ou sinon de mettre sa fréquence à jour à chaque nouvelle occurrence. La classe « *OccurObjet* » est fournie à l'extrait de code 4.4. C'est donc l'objet de type « *OccurObjet* » qui emmagasine et gère les cooccurrences pour chaque lemme.

Dans la première phrase, on ne retrouve aucun nom autre que « voiture » lui-même. La table de hachage « Noms » reste donc vide pour l'instant. On retrouve en revanche dans cette phrase un adjectif, « rouge ». On crée donc un élément dans la table de hachage « Adjectifs » dont la clé est « rouge » et la fréquence est « 1 », car c'est la première fois que l'adjectif « rouge » apparaît comme cooccurrence avec le lemme « voiture ». Nous procédons de la même façon pour le verbe « rouler » et l'adverbe « rapidement », comme on peut le voir à la Figure 4.8.

Extrait de code 4.4 : Objet (classe) pour les cooccurrences

```
public class OccurObjet {

HashMap<String,Integer> noms      = new HashMap<>();
HashMap<String,Integer> verbes    = new HashMap<>();
HashMap<String,Integer> adjectifs = new HashMap<>();
HashMap<String,Integer> adverbes  = new HashMap<>();

// ---------- METHODES SET -------------------

public void SetNoms (String mot){
ajouteMot(noms,mot);
}
public void SetVerbes (String mot){
ajouteMot(verbes,mot);
}
public void SetAdjectifs (String mot){
ajouteMot(adjectifs,mot);
}
public void SetAdverbes (String mot){
ajouteMot(adverbes,mot);
}

public HashMap<String,Integer> GetNoms(){
HashMap<String,Integer> table = noms;
return table;
}
public HashMap<String,Integer> GetVerbes(){
HashMap<String,Integer> table = verbes;
return table;
}
public HashMap<String,Integer> GetAdjectifs(){
HashMap<String,Integer> table = adjectifs;
return table;
}
public HashMap<String,Integer> GetAdverbes(){
HashMap<String,Integer> table = adverbes;
return table;
}

public void ajouteMot(HashMap<String,Integer> table, String mot){
Integer freqCourante;
boolean existe=table.containsKey(mot);
if(!existe){
    table.put(mot, 1);}
else{
    freqCourante=table.get(mot);
    table.replace(mot,freqCourante+1);}
}
}
```

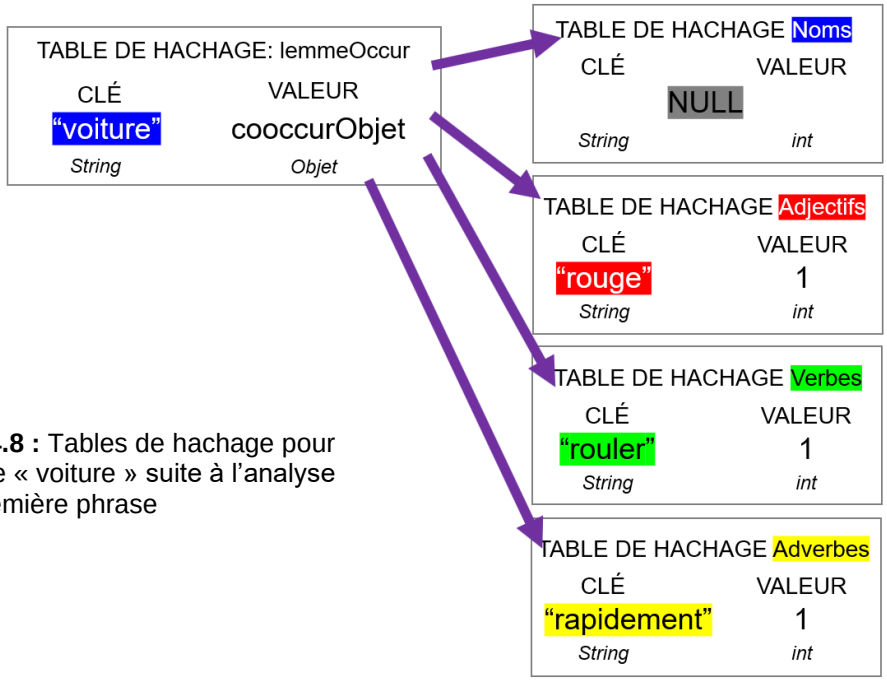


Figure 4.8 : Tables de hachage pour le lemme « voiture » suite à l'analyse de la première phrase

Nous procédons ainsi encore de la même façon pour le lemme « rouge », le deuxième que l'on retrouve à la Phrase 1. Le résultat apparaît à la Figure 4.9.

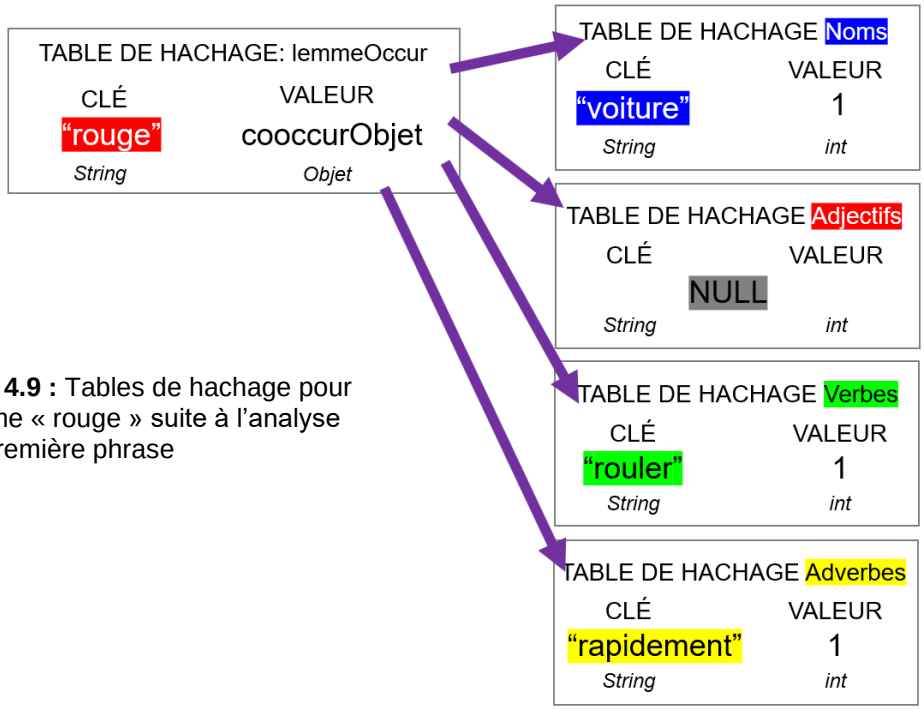


Figure 4.9 : Tables de hachage pour le lemme « rouge » suite à l'analyse de la première phrase

Et ainsi de suite, pour les deux autres lemmes de la Phrase 1 (voir Figures 4.10 et 4.11).

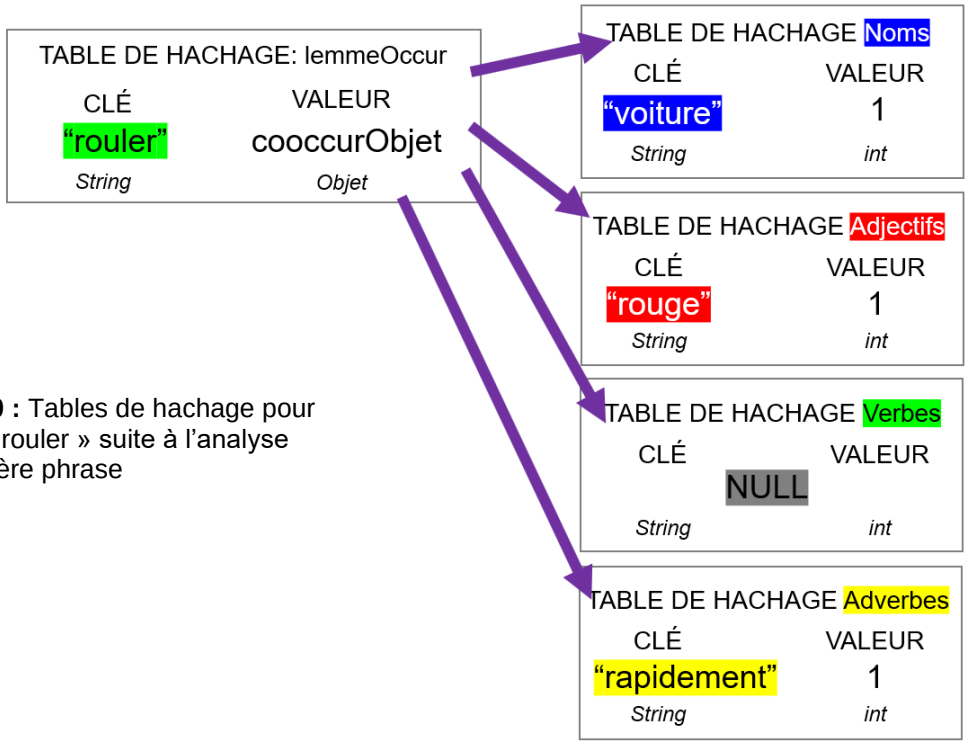


Figure 4.10 : Tables de hachage pour le lemme « rouler » suite à l'analyse de la première phrase

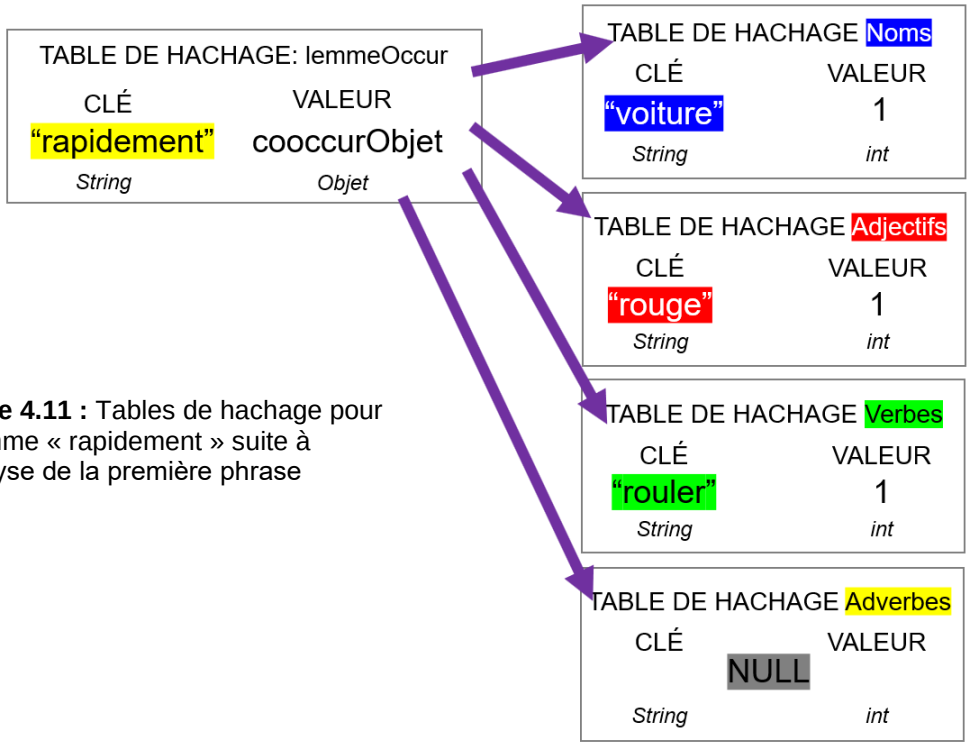


Figure 4.11 : Tables de hachage pour le lemme « rapidement » suite à l'analyse de la première phrase

Suivant l'analyse de cette première phrase, la table de hachage « *lemmeOccur* » comporte donc quatre éléments, correspondant aux quatre mots de la phrase appartenant aux classes de mots nous intéressant.

Continuons le processus avec la deuxième phrase, qui contient les lemmes « voiture », « passer », « lumière » et « rouge ». Commençons avec le lemme « voiture ». Tout d'abord, on constate que la table « *lemmeOccur* » comporte déjà une clé « voiture ». Il ne faut donc pas ici créer un nouvel élément, mais plutôt le mettre à jour avec les données de cette deuxième occurrence du lemme « voiture ». L'élément mis à jour de la table de hachage « *lemmeOccur* » correspondant à la clé « voiture » est illustré à la Figure 4.12.

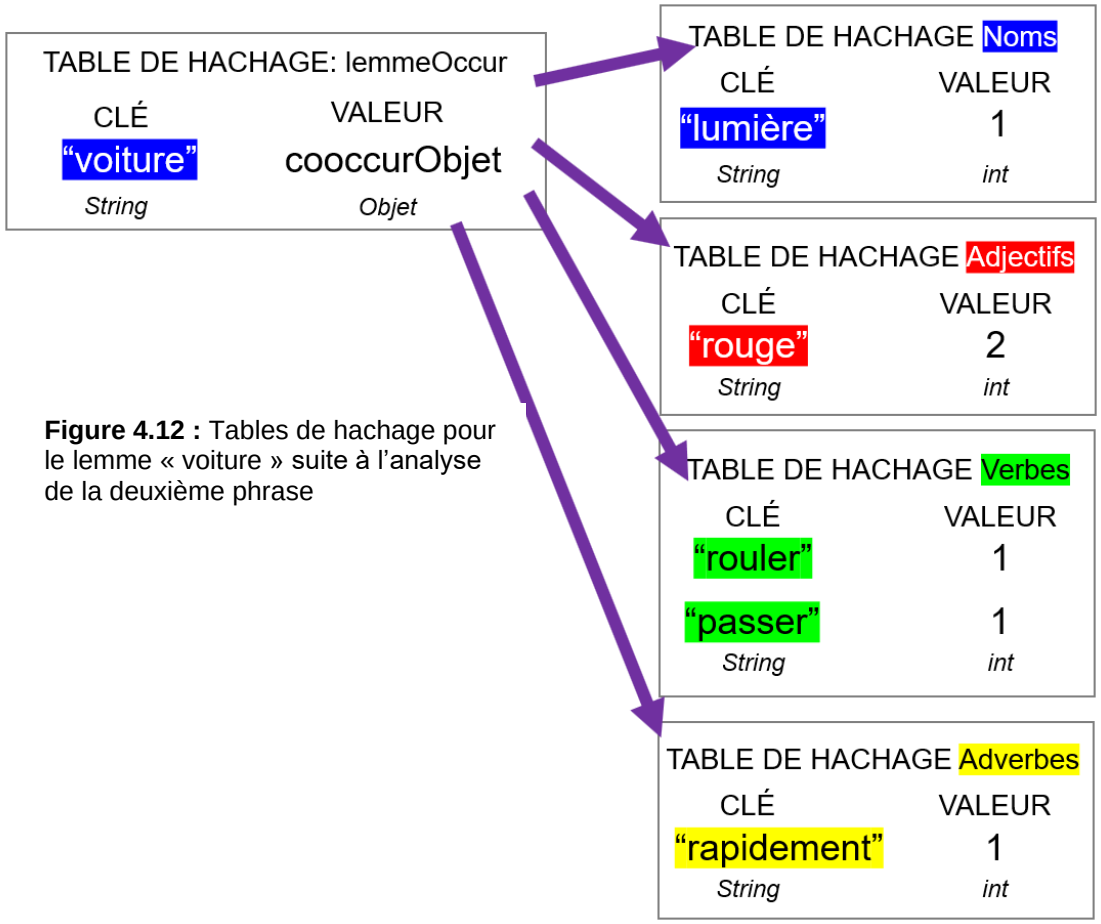


Figure 4.12 : Tables de hachage pour le lemme « voiture » suite à l'analyse de la deuxième phrase

Nous constatons maintenant que la table de hachage « Noms » n'est plus vide, car le nom « lumière » est une cooccurrence du lemme « voiture ». Nous constatons aussi que bien que la table « Adjectifs » ne comporte toujours que la clé « rouge », la valeur associée est maintenant « 2 », car il y a maintenant deux situations où les lemmes « voiture » et « rouge » se retrouvent dans la même phrase. Dans la table de hachage « Verbes » pour le lemme « voiture », il a maintenant fallu ajouter le verbe « passer » avec une fréquence de « 1 », puisque c'est la première fois que ces deux lemmes se retrouvent dans la même phrase. La table de hachage « Verbes » se retrouve donc avec un élément de plus. Finalement, la table de hachage « Adverbes » n'est pas mise à jour, car la deuxième phrase ne contient aucun adverbe. Nous poursuivons ensuite l'analyse avec le lemme « passer » (Figure 4.13).

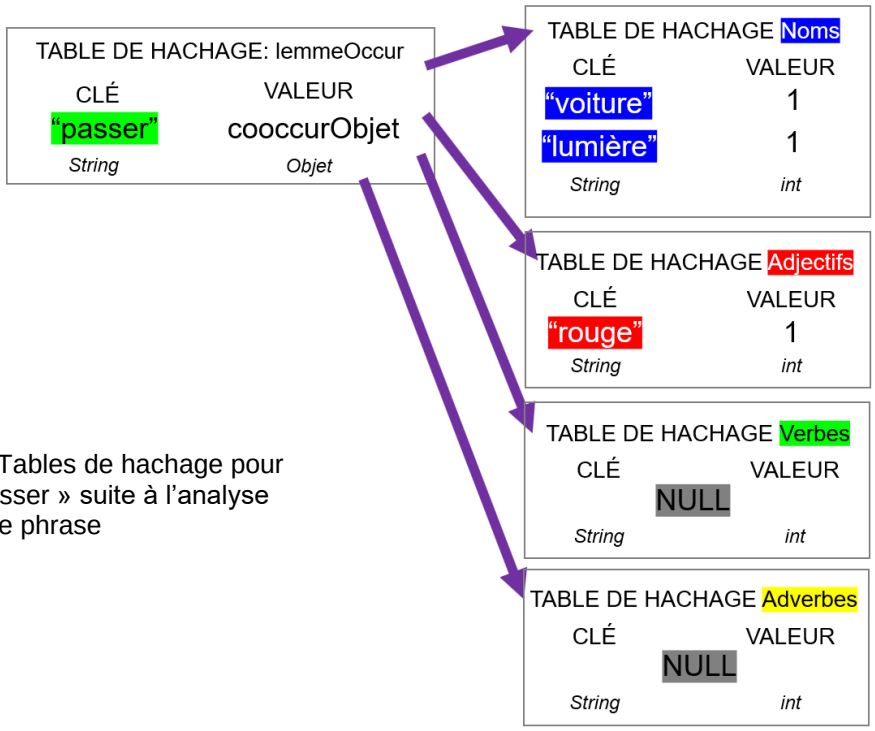


Figure 4.13 : Tables de hachage pour le lemme « passer » suite à l'analyse de la deuxième phrase

Comme la table de hachage « *lemmeOccur* » ne contenait pas déjà la clé « passer », il a fallu y ajouter un élément. L'objet correspondant à la clé « passer » comporte toujours les quatre tables de hachage correspondant aux quatre classes de mots, mais deux d'entre elles demeurent vides, car aucun adverbe ni aucun autre verbe ne se retrouve dans la phrase où se trouve le lemme « passer ». En revanche, on y retrouve deux noms (« voiture » et « lumière »), et un adjectif (« rouge »). Nous poursuivons ensuite avec le lemme « lumière » (Figure 4.14).

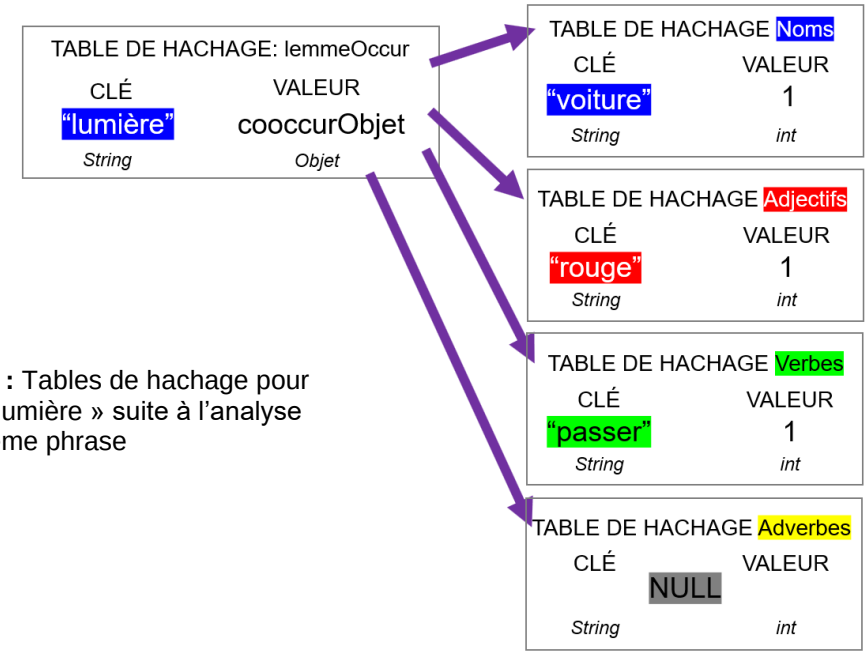


Figure 4.14 : Tables de hachage pour le lemme « lumière » suite à l'analyse de la deuxième phrase

Tout comme pour le lemme « passer », il a fallu créer un nouvel élément dans la table de hachage « *lemmeOccur* », auquel sont à nouveau associées quatre tables de hachage (Figure 4.14). On ne retrouve aucun adverbe pour l'instant.

On termine ensuite avec le lemme « rouge ». On constate que la clé « rouge » existe déjà dans la table de hachage « *lemmeOccur* ». Il ne faut donc pas créer de nouvel élément, mais plutôt mettre à jour l'élément existant (Figure 4.15) :

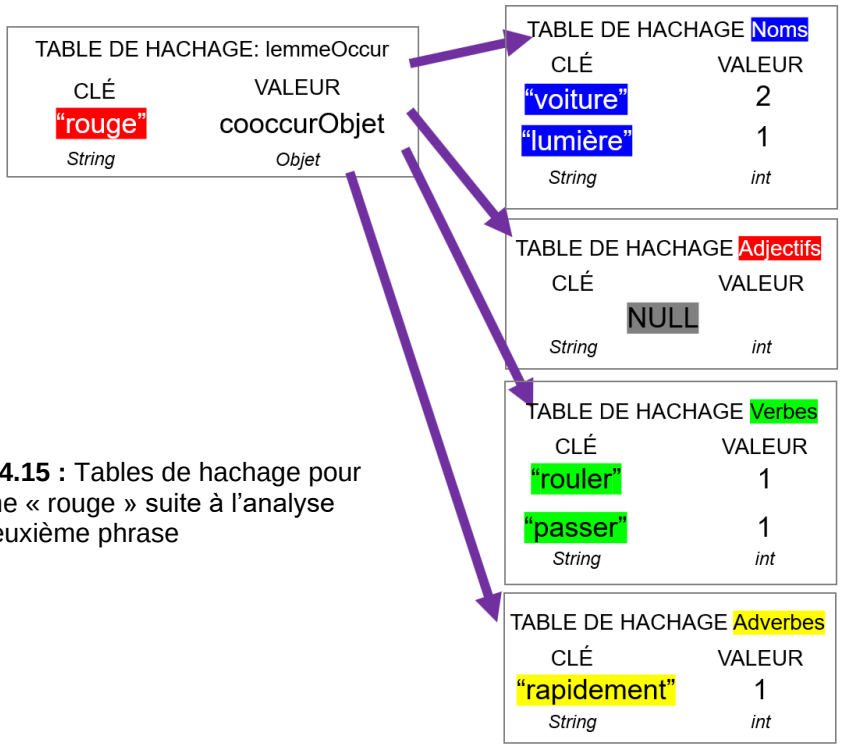


Figure 4.15 : Tables de hachage pour le lemme « rouge » suite à l'analyse de la deuxième phrase

Pour la table de hachage « Noms », on retrouve le nom « voiture » avec une fréquence de « 2 », puisque la cooccurrence « voiture-rouge » se retrouve dans les deux phrases analysées jusqu'à présent. En revanche, la fréquence pour le nom « lumière » n'est que « 1 ». Nous constatons qu'aucun autre adjectif ne s'est retrouvé dans les deux phrases ayant contenu le lemme « rouge ». On retrouve cependant deux verbes, avec une fréquence de « 1 » dans les deux cas, et un seul adverbe.

L'analyse est maintenant terminée et toutes les tables de hachages ont été construites. La table de hachage « *lemmeOccur* » comporte six éléments, tous distincts (« voiture », « rouge », « rouler », « rapidement », « passer » et « lumière »). Par définition, chaque valeur correspondant aux clés de *lemmeOccur* est un objet comportant quatre tables de hachage. Le contenu de ces tables de hachage varie en fonction des cooccurrences, et la version finale de ces tables, après l'analyse des deux phrases, a été illustrée aux Figures 4.12 à 4.15.

Il est toutefois pertinent de mentionner que l'algorithme de cooccurrences suggéré ici est coupable de « dédoublement » d'information, car si par exemple les lemmes « rouge » et « voiture » sont cooccurrents, on enregistrera cette information à deux endroits, c'est-à-dire sous la clé « rouge » et aussi sous la clé « voiture » dans la table « *lemmeOccur* ». Il en résulte donc un plus grand usage de l'espace mémoire. Cependant, l'enregistrement en double de cette information permet

par la suite un accès plus rapide lors de la création de phrases automatiquement lemmatisées à l'Étape 2.

Il faut en effet se rappeler que le but de la création de ces tables de cooccurrences est d'aider à former des phrases ayant potentiellement un minimum de sens, lors de l'Étape 2 du projet. La structure choisie pour noter les cooccurrences (tables de hachages dans un objet lui-même contenu dans une table de hachage) l'a été en fonction de la facilité d'accès à l'information, une fois celle-ci créée. En effet, si on sélectionne par exemple le verbe « rouler » pour créer une phrase à l'Étape 2, on n'aura qu'à chercher la clé « rouler » dans la table de hachage « *lemmeOccur* ». Si nous recherchons un nom pour compléter notre phrase comprenant le lemme « rouler », on cherchera dans la table de hachage « Noms » correspondant au lemme « rouler » les noms possibles. Les fréquences incluses pour chaque nom cooccurrent nous permettront d'utiliser le concept de probabilités pour sélectionner au hasard le nom à inclure. L'information recherchée est donc facilement et rapidement accessible.

Dans l'algorithme illustré ici pour les cooccurrences, nous n'avons considéré que les cooccurrences à l'intérieur *d'une seule et même phrase*. L'algorithme aurait pu être modifié pour, par exemple, noter les cooccurrences en considérant, en plus de la phrase courante, les phrases précédente et suivante, ou même davantage. Ceci aurait permis de tenir compte du fait que dans un texte, les phrases successives portent généralement sur les mêmes idées et sujets.

4.4.1. Homographes et tables de cooccurrences

L'outil de lemmatisation développé pour l'Étape 1 du projet énumère les homographes possibles pour un mot donné. Par exemple, pour le mot « montres », deux lemmes possibles seront donnés, soit le verbe « montrer » ou soit le nom commun « montre ». Comme l'algorithme de lemmatisation de base (avant la désambiguïsation) ne permet pas de déterminer lequel de ces deux homographes est celui réellement utilisé dans le texte, l'algorithme de cooccurrences décrit ici inclut dans ce cas les deux lemmes, soit « montrer » et « montre ». Prenons l'exemple suivant :

Phrase : L'homme regarde distraitement sa belle montre.

On en extrait les mots appartenant uniquement aux quatre classes de mots qui nous intéressent (en utilisant le même code de couleurs que précédemment):

Phrase : L'homme regarde distraitement sa belle montre.

On retire les autres mots :

homme regarde distraitement belle montre

On remplace les mots par leurs lemmes respectifs :

Lemmes : homme regarder distraitement beau montre montrer

On note ici que pour le mot « montre », nous avons inclus deux lemmes possibles, soit « montre » et « montrer », puisqu'on considère ici le cas où la désambiguïsation des homographes n'a pas encore eu lieu. L'algorithme de cooccurrences doit donc considérer ces deux possibilités. Ainsi, il bâtit des tables de hachage pour l'élément « montre » ainsi que pour l'élément « montrer », qui sont identiques, s'il ne s'agit que de cette seule phrase. Cependant, nous devons nous assurer de ne pas considérer le nom « montre » comme étant une cooccurrence du verbe « montrer » et vice-versa. Mais lorsqu'on effectue la désambiguïsation (décrite en détail à la Section 4.6) avant d'effectuer l'analyse des cooccurrences, ce problème ne se pose pas. Cette situation illustre une fois de plus la pertinence d'effectuer la désambiguïsation des homographes dans le cadre de ce travail.

Comme autre exemple en lien avec l'homographie, on peut faire face à deux mots distincts, mais correspondant au même lemme. Citons le nom masculin pluriel « dîners » (« C'est lui qui prépare les dîners ») et le verbe conjugué « dîneront » (« Ils dîneront plus tard, car le repas n'est pas encore prêt »). Une fois lemmatisés, ces deux mots se réduiront au lemme identique « dîner ». C'est donc dire que l'algorithme de cooccurrences ne tiendra pas compte du fait que des mots avec différentes classes grammaticales (ici un nom commun et un verbe) se retrouveront à être regroupés sous un même lemme. Il s'agit d'une faiblesse, que l'outil de désambiguïsation actuel ne permet pas de régler. Cette faiblesse pourrait toutefois être évitée en créant quatre tables de hachage du type « *lemmeOccur* », soit une pour chaque classe de mots nous intéressant – verbes, noms, adjectifs et adverbe. Mais nous ne procéderons pas ainsi pour ce projet, par simplicité.

4.5. Répertorier les homographes

Avant d'appliquer l'algorithme de désambiguïsation, il est possible de faire l'inventaire de tous les homographes présents dans le corpus de référence. Cette étape est pertinente car elle permet de déterminer la proportion d'homographes appartenant à des classes grammaticales différentes, sachant que ceux-ci sont plus faciles à désambiguïser. En particulier, cette analyse permet de constater quelles classes grammaticales entrent généralement en jeu pour ces homographes, ce qui peut par la suite guider la sélection d'indices syntaxiques permettant de distinguer les classes grammaticales entre elles.

En sortie de l'Étape 1, trois types d'information en lien avec la fréquence des homographes sont fournis. Dans un premier lieu, un tableau fournit les fréquences absolues des homographes dans le corpus de référence, en fonction des classes grammaticales impliquées. Ce tableau comporte neuf lignes et neuf colonnes, correspondant aux neuf grandes classes grammaticales identifiées pour ce projet et listées au Tableau 3.1. Ainsi, le mot « pas », qui peut être ou bien un nom commun ou bien un adverbe, est comptabilisé dans ce tableau à deux endroits : à la ligne « nom » et colonne « adverbe », et à la ligne « adverbe » et colonne « nom ». Le mot « fils » quant à lui, comporte deux possibilités appartenant à la même classe grammaticale (nom commun singulier « fils » signifiant un enfant de sexe masculin, et nom commun pluriel « fils » comme dans « fils électriques »). Le mot « fils » est donc comptabilisé dans une seule cellule du tableau, correspondant à la ligne et à la colonne pour « nom commun ». Un exemple de tableau d'homographes vide est illustré au Tableau 4.10. Le tableau qui en résulte est l'équivalent d'une matrice symétrique, car à chaque case verte correspond une case rose. Les cases bleu foncé correspondent quant à eux aux homographes appartenant à la même classe grammaticale.

Si par exemple on observe 15 occurrences de l'homographe « pas » dans le corpus, cette valeur de 15 s'ajoute à toutes les autres occurrences d'homographes de type « nom-adverbe » au Tableau 4.10 (deux cases). Cependant, comme deuxième type d'information, on peut se limiter aux formes distinctes d'homographes. Dans un tel cas, l'homographe « pas », indépendamment du nombre de fois qu'on le retrouve dans le corpus, ne contribuerait que pour une valeur de 1 dans le tableau. Ces deux tableaux (valeurs totales et valeurs distinctes) sont fournis en sortie de l'algorithme pour le corpus de référence.

Tableau 4.10 : Tableau d'homographes pour y répertorier soit les fréquences absolues, ou les fréquences d'homographes distincts

	Verbe	Adjectif	Nom	Adverbe	Déterminant	Pronom	Préposition	Conjonction	Interjection
Verbe									
Adjectif									
Nom									
Adverbe									
Déterminant									
Pronom									
Préposition									
Conjonction									
Interjection									

Finalement, l'algorithme fournit aussi en sortie la liste détaillée de chaque homographe distinct, les classes grammaticales qui y sont associées, et son nombre d'occurrence dans le corpus. Pour le mot « pas » par exemple, on pourrait retrouver :

```
pas      15    3     4
```

Ceci nous indique qu'on retrouve l'homographe « pas » 15 fois dans le corpus, et qu'il appartient aux classes grammaticales associées aux codes 3 (nom) et 4 (adverbe). Cette information détaillée par homographe permet d'identifier des cas particuliers pouvant potentiellement causer problème. Une telle identification permet alors, au besoin, de mettre au point des stratégies particulières pour désambiguïser ces homographes.

4.6. Désambiguïsation des homographes par apprentissage machine

Le but de l'algorithme de désambiguïsation développé pour ce projet est de déterminer la classe grammaticale de chaque homographe, parmi les neuf possibilités décrites au Tableau 3.1. On ne cherche donc pas ici à distinguer les homographes issus d'une même classe grammaticale (« suis » du verbe être, et « suis » du verbe suivre par exemple). Un tel défi est hors de la portée de ce projet. On ne cherche pas directement non plus par cet algorithme à accoler une étiquette plus précise à chaque mot désambiguïsé. Toujours est-il que celle-ci deviendra disponible une fois la classe grammaticale identifiée, par l'entremise de l'information emmagasinée dans la table de hachage « *tableRef* ». En somme, on ne cherche en effet qu'à associer chaque homographe à un chiffre de 1 à 9, correspondant aux codes de classes grammaticales du Tableau 3.1.

L'approche générale adoptée ici pour la désambiguïsation du corpus de référence est l'apprentissage machine, aussi appelé « apprentissage automatique ». L'algorithme qui nous intéresse plus particulièrement en est un de « classification ». Par ce processus, un certain nombre de caractéristiques (appelées « *features* » en anglais dans ce contexte) sont évaluées pour chaque homographe, la plupart du temps en fonction des autres mots présents dans la phrase. Ces caractéristiques sont quantifiées, donc exprimées numériquement. Dans la plupart des cas, une valeur de 0 ou de 1 est associée à chaque caractéristique, selon si elle est observée ou non. Certaines des caractéristiques choisies pour ce projet ont été présentées aux Tableaux 3.13 et 3.14 du Chapitre 3.

L'apprentissage machine consiste généralement à évaluer ces caractéristiques d'abord pour des cas où un étiquetage a été effectué « manuellement ». C'est l'étape qu'on appelle l'entrainement. On indique à l'algorithme, pour un certain nombre de mots, la classe grammaticale à laquelle ils

appartiennent et on fournit les caractéristiques qu'on a observées pour ces mots. On permet ainsi à l'algorithme d'associer les valeurs des caractéristiques aux neuf classes grammaticales possibles du Tableau 3.1.

Comme deuxième étape, on effectue ce qu'on appelle le « test » ou « évaluation ». À cette étape, la classe grammaticale n'est pas connue. C'est ce qu'on essaie d'évaluer en comparant les caractéristiques du mot sous analyse avec celles des mots déjà étiquetés lors de l'entrainement. Cette deuxième étape peut servir d'outil pour quantifier la performance du modèle développé sur la base des données d'entrainement. Mais surtout, c'est à cette deuxième étape que l'apprentissage machine prend tout son sens, car on peut alors l'appliquer pour classifier de nouvelles observations. Autrement dit, on peut alors se servir du modèle pour classifier les homographes de *nouveaux* textes, c'est-à-dire des textes qui n'ont pas servi à entrainer le modèle.

4.6.1. Caractéristiques syntaxiques (« *features* »)

Tout algorithme d'apprentissage machine doit être alimenté par des observations consistant à évaluer chaque mot utilisé en fonction de certaines caractéristiques. Dans le cas de l'apprentissage machine *supervisé*, comme c'est le cas ici, ces caractéristiques doivent être déterminées *a priori* en fonction de leur capacité à faciliter le processus de classification. Certaines des caractéristiques employées par l'algorithme de désambiguïsation ont été présentées aux Tableaux 3.13 et 3.14. Ces caractéristiques ont donc été soigneusement choisies car on considère qu'elles fourniront les indices nécessaires visant à déterminer la classe grammaticale de chaque mot. Il n'est pas nécessaire ici d'élaborer davantage sur la pertinence des caractéristiques sélectionnées, cette discussion ayant été faite au Chapitre 3.

Au moment de l'entrainement du modèle, on assigne d'abord à chaque mot la classe grammaticale à laquelle il appartient dans le contexte de la phrase d'où il est extrait. On évalue ensuite toutes les caractéristiques de ce mot, toujours dans ce même contexte de la phrase. On a choisi ici de combiner l'information portant sur tous les mots étudiés du corpus de référence, dans un fichier texte qui servira par la suite à déterminer les paramètres du modèle d'apprentissage machine. Le Tableau 4.11 donne un exemple fictif de ce à quoi un tel tableau peut ressembler, où seules quelques caractéristiques ont été incluses, par souci de concision. Une discussion plus approfondie de l'évaluation des caractéristiques aura lieu dans les sections portant sur l'entrainement du modèle.

Tableau 4.11 : Extrait du tableau des caractéristiques généré lors de l'entrainement, avec valeurs fictives. Seules quelques colonnes et quelques lignes affichées. Ce tableau, sans l'entête, est enregistré dans un fichier texte (ASCII)

Classe grammaticale	Caractéristique 1	Caractéristique 2	Caractéristique 3	Caractéristique 4	Caractéristique 5
5	0	1	0	0	1
2	1	1	0	0	0
3	0	0	0	1	0
1	0	1	0	1	0
5	0	0	0	1	1
3	1	1	0	0	0

4.6.2. Techniques d'apprentissage machine

Il existe plusieurs algorithmes d'apprentissage machine permettant de classifier des observations selon des catégories prédéfinies, de façon systématique et efficace. De tels algorithmes sont tout à fait généraux; on peut les appliquer à toutes sortes de problèmes de classification dans des domaines variés. Ce qui importe ici est que ces algorithmes se prêtent bien à la classification d'homographes selon leur classe grammaticale, une tâche critique de tout outil de lemmatisation de texte. Deux de ces types d'algorithmes généraux sont présentés à la section suivante : les arbres de décision et la régression logistique binaire.

4.6.2.1. Arbres de décision

Les arbres de décision se prêtent bien de façon générale à des processus de classification comme celui dont il est question ici impliquant les classes grammaticales. Tel que mentionné dans Wikipedia (2023b), « un arbre de décision est un outil d'aide à la décision représentant un ensemble de choix sous la forme graphique d'un arbre. Les différentes décisions possibles sont situées aux extrémités des branches (les « feuilles » de l'arbre), et sont atteintes en fonction de décisions prises à chaque étape. » Un arbre de décision permet de considérer plusieurs classes à la fois au besoin, et il est facile à incorporer à un programme informatique, grâce à une série d'énoncés conditionnels (« IF », et « ELSE »). Il existe divers algorithmes (ID3, CART, etc.) pour optimiser ces arbres, ce qui implique de sélectionner les caractéristiques servant à départager les classes, et de déterminer les valeurs-seuil devant être utilisées pour chacune des caractéristiques.

Dans le contexte actuel, un arbre de décision « global » pourrait par exemple être mis en place, sur la base de caractéristiques (Section 4.6.1) telles que celles démontrées aux Tableaux 3.13 et 3.14. L'algorithme, utilisant ces données en entrée, pourrait alors prédire la classe grammaticale de chaque mot fourni en entrée. Un tel arbre doit être suffisamment « profond » pour s'assurer que les neuf possibilités de classes grammaticales en émergent (Figure 4.16).

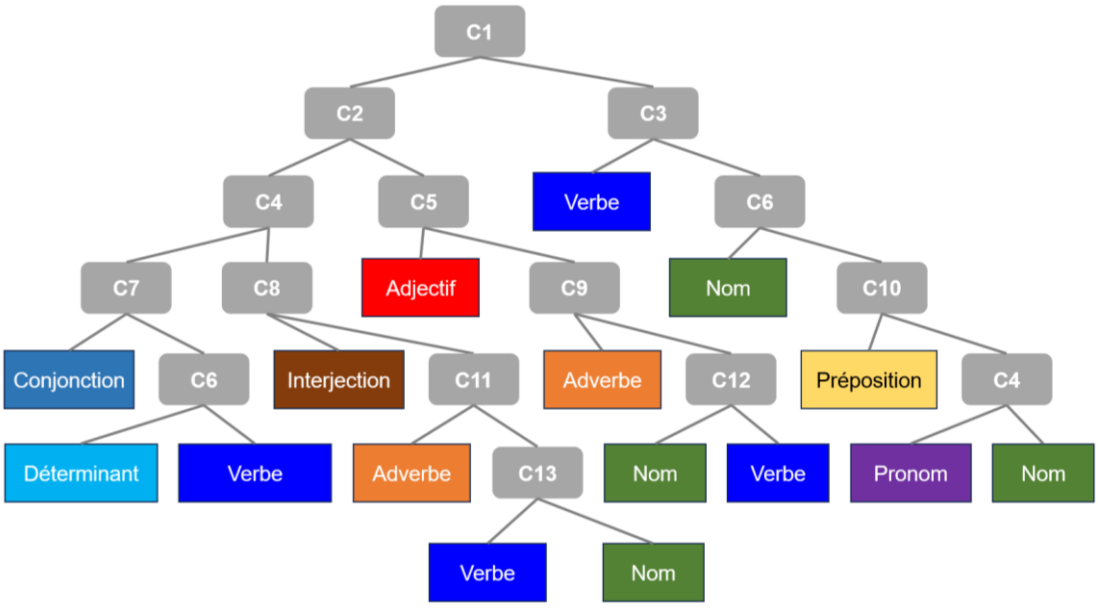


Figure 4.16 : Illustration d'un arbre de décision fictif générant des prédictions pour les neuf classes grammaticales. L'arbre doit être suffisamment profond pour que les neuf options en émergent. Les caractéristiques sont représentées par C1, C2, etc. Note : la même caractéristique et la même classe grammaticale peuvent apparaître plus d'une fois

Tableau 4.12 : Tableau indiquant toutes les paires possibles de classes grammaticales. Les chiffres dans les cases ne servent qu'à faciliter le décompte (36 paires possibles)

Adjectif	Nom	Adverbe	Déterminant	Pronom	Préposition	Conjonction	Interjection	
1	2	3	4	5	6	7	8	Verbe
	9	10	11	12	13	14	15	Adjectif
		16	17	18	19	20	21	Nom
			22	23	24	25	26	Adverbe
				27	28	29	30	Déterminant
					31	32	33	Pronom
						34	35	Préposition
							36	Conjonction

Toutefois, il appert qu'aucun mot de la langue française n'est un homographe pouvant se retrouver dans n'importe laquelle des neuf classes grammaticales décrites au Tableau 3.1. Pour la très grande majorité d'entre eux, les homographes n'offrent que deux options de classes. En effet, comme on le constatera au Chapitre 5, près de 93%[8] des homographes issus des corpus de références choisis, n'impliquent que deux classes distinctes. Par exemple, le mot « demande » peut être soit un nom commun, soit un verbe, mais pas un adverbe, ni un pronom, ni un déterminant, etc.

Si on se concentre sur les homographes n'appartenant qu'à deux classes grammaticales, on peut donc simplifier l'analyse en comparant les probabilités par paires de classes. Par exemple, pour le mot « demande », on veut déterminer si ce mot est un verbe ou un nom commun, sans avoir à se soucier des autres classes. On doit donc bâtir, pour chaque paire possible de classes grammaticales, des sous-ensembles de mots se rapportant uniquement à chaque classe incluse dans la paire. Le Tableau 4.12 illustre toutes les paires de classes grammaticales possibles, considérant le total de neuf classes. À partir du Tableau 4.12, on constate donc qu'il y a 36 paires possibles de classes grammaticales obtenues sur la base d'un total de neuf classes. En fait, cette valeur de 36 découle d'un calcul de probabilités combinatoires :

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} = \frac{2!}{9!(9-2)!} = 36 \tag{4.1}$$

A l'Équation 4.1, le paramètre n correspond au nombre de classes choisies par groupe, donc une valeur de 2 si on ne considère que des paires, et k est le nombre total de classes, soit 9. Quand on se contente ainsi de ne considérer que les *paires* de classes grammaticales, on se retrouve à devoir bâtir 36 fichiers de caractéristiques correspondant à chacune de ces paires. On obtient par exemple un fichier de données ne contenant que des adjectifs et des noms (codes « 2 » et « 3 »), un autre ne contenant que des adjectifs et des verbes, et ainsi de suite. Le Tableau 4.13 illustre ce à quoi peut ressembler chacun de ces « sous-tableaux » de caractéristiques. On constate en effet, au Tableau 4.13, que les seules valeurs apparaissant dans la colonne des classes grammaticales sont ici le « 1 » et le « 2 », correspondant aux verbes (« 1 ») et aux adjectifs (« 2 »). Chacun des 36 fichiers générés n'inclut donc ainsi que les mots appartenant à deux classes grammaticales, contrairement au Tableau 4.11 qui peut contenir des exemples de toutes les classes. Ces 36 sous-tableaux permettent de générer 36 arbres de décision *spécialisés* pour chaque paire de classes grammaticales. Ces 36 arbres ont le potentiel d'offrir une meilleure

[8] La valeur de 93% est basée sur la fréquence d'apparition des homographes, et non sur le nombre de formes distinctes d'homographes

classification qu'un arbre global et unique considérant en simultané les neuf classes grammaticales. On aboutit par exemple avec un arbre comparant uniquement les verbes aux noms communs, un autre arbre comparant les adjectifs aux verbes, et ainsi de suite. La Figure 4.17 illustre un arbre de décision fictif n'impliquant que deux classes grammaticales.

Mais tel que mentionné plus haut, certains homographes se rapportent à trois classes grammaticales au lieu de deux. En effet, environ 5% des homographes issus des corpus de référence choisis, offrent trois options de classes grammaticales. Par exemple, le mot « ferme » dont il a été question plus tôt, peut être un nom commun, un adjectif ou un verbe. L'arbre de décision fictif illustré à la Figure 4.18 est un exemple d'arbre permettant de déterminer si un homographe est un nom commun, un adjectif ou un verbe, comme pour le mot « ferme ».

Mais plutôt que de devoir créer 36 arbres de décision spécialisés, il faut maintenant en créer 84, afin de générer toutes les combinaisons possibles impliquant trois classes distinctes. Cette valeur de 84 est obtenue en utilisant à nouveau l'Équation 4.1 portant sur les probabilités combinatoires, mais cette fois en utilisant la valeur du paramètre $n = 3$, comme on peut le voir à l'Équation 4.2.

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} = \frac{3!}{9!(9-3)!} = 84 \tag{4.2}$$

Tableau 4.13 : Exemple fictif d'un des 36 tableaux de caractéristiques généré à l'entrainement, se rapportant ici uniquement aux verbes (code=1) et aux adjectifs (code=2). Seules quelques colonnes et quelques lignes affichées. Ce tableau, sans l'entête, est enregistré dans un fichier texte

Classe grammaticale	Caractéristique 1	Caractéristique 2	Caractéristique 3	Caractéristique 4	Caractéristique 5
1	0	1	0	0	1
2	1	1	0	0	0
1	0	0	0	1	0
1	0	1	0	1	0
1	0	0	0	1	1

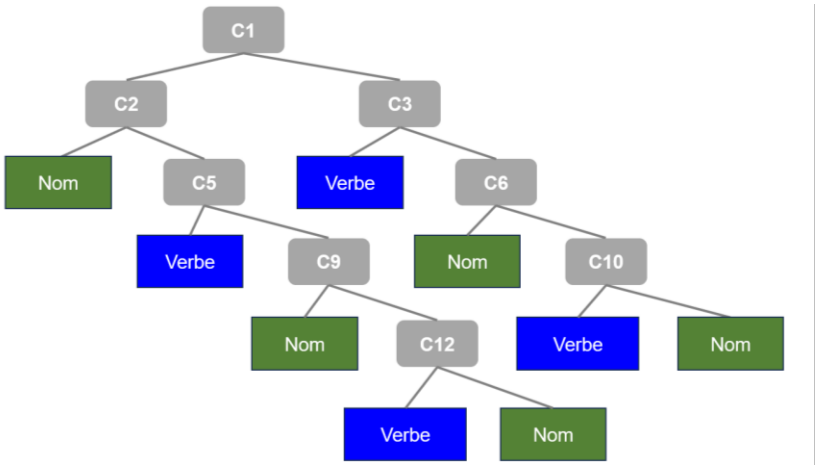


Figure 4.17 : Illustration d'un arbre de décision fictif générant des prédictions pour les homographes de type « verbe-nom ». En tout, 36 arbres comme celui-ci sont nécessaires pour toutes les paires possibles des neuf classes grammaticales

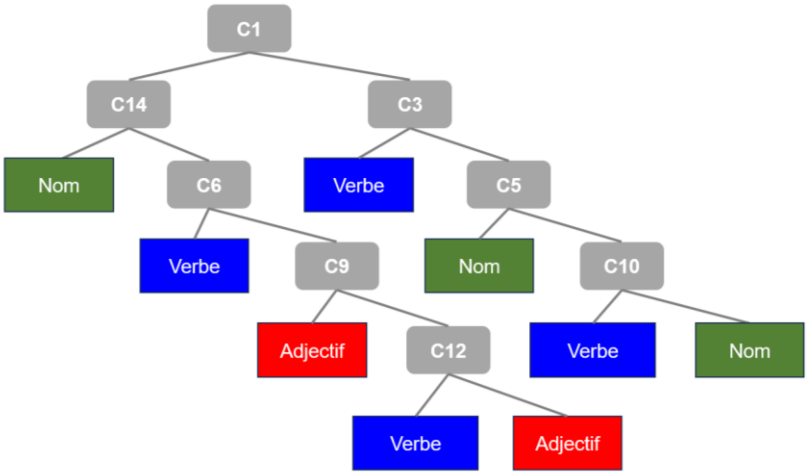


Figure 4.18 : Illustration d'un arbre de décision fictif générant des prédictions pour des homographes de type « verbe-nom-adjectif », comme le mot « ferme ». En tout, 84 arbres comme celui-ci sont nécessaires pour tous les trios possibles de classes grammaticales

Mais on peut heureusement se simplifier la tâche en se limitant aux seuls arbres pour lesquels des homographes ont été recensés dans le corpus de référence, les autres étant inutiles. En effet, on ne retrouve parmi les homographes issus des corpus de référence choisis que cinq combinaisons distinctes de trois classes grammaticales :

- Verbe, adjectif, nom (exemples : « vivant », « fait », « vue »)
- Verbe, adjectif, conjonction (seul exemple : « pourvu »)
- Verbe, nom, préposition (seul exemple : « devant »)
- Adjectif, nom, adverbe (exemples : « fort », « droit », « proche »)
- Adjectif, nom, pronom (seuls exemples : « certains », « certaines »)

Ainsi, plutôt que de devoir générer 84 arbres de décision pour les homographes comportant trois possibilités de classes grammaticales, on pourrait se contenter d'en créer uniquement cinq. Mais il faut par la suite tenir compte du fait que certains mots peuvent correspondre à plus de trois classes grammaticales possibles. Au sein des corpus de référence, on retrouve par exemple les deux mots suivants :

- « tout » : ce mot peut être un adjectif, un nom commun, un adverbe, un déterminant ou un pronom (donc cinq possibilités)
- « point » : ce mot peut-être un verbe (participe passé du verbe « poindre »), un participe passé employé seul (donc considéré comme un adjectif), un nom commun, ou un adverbe (donc quatre possibilités)

Des arbres de décision devraient donc être construits aussi pour ces deux possibilités. Il existe peut-être d'autres homographes dans la langue française comportant plus de trois classes grammaticales possibles, mais aucun autre n'a été identifié parmi les homographes issus des corpus de référence sélectionnés.

Les arbres de décision sont avantageux à plusieurs niveaux. D'une part, ils sont intuitifs, car faciles à interpréter par un humain d'un seul coup d'œil. De plus, ils permettent de facilement identifier les quelques caractéristiques les plus pertinentes, soit celles qu'on retrouve le plus près de la racine de l'arbre. De plus, une fois développés par un algorithme tel que « ID3 » ou

« CART », les arbres de décision s'incorporent facilement à un outil de lemmatisation, car il ne suffit que d'inclure une série d'énoncés du type « IF-ELSE ».

Mais les arbres de décision comportent leur lot de désavantages. Par exemple, à mesure que de nouvelles observations viennent alimenter l'algorithme de création de l'arbre, il se peut que sa structure change. En effet, l'ordre des tests des caractéristiques, ou même la sélection des caractéristiques présentes dans l'arbre peut changer, ce qui implique des modifications manuelles à la programmation de l'arbre dans l'outil de lemmatisation. Mais surtout, les arbres de décision, typiquement, ne font intervenir que quelques-unes des caractéristiques, ignorant les autres, même si ces dernières peuvent en théorie aussi contribuer à affiner la décision. De plus, bien que des arbres plus « profonds » offrent de meilleurs résultats sur les données d'entrainement, ils engendrent souvent un surajustement (« *overfitting* » en anglais) sur les données de test. Autrement dit, l'arbre devient « surspécialisé » pour les données d'entrainement, ce qui se retrouve à diminuer sa performance lors de la désambiguïsation. Un désavantage additionnel est que les prédictions faites par les arbres de décision ne sont pas accompagnées d'une probabilité, comme c'est le cas avec la régression logistique binaire, comme on le verra plus loin. Finalement, bien qu'on n'en discute pas ici, les résultats préliminaires obtenus avec des arbres de décision se sont avérés moins efficaces qu'avec la régression logistique. Pour ce projet, c'est donc l'approche de la régression logistique qui a été favorisée. Aucune autre discussion portant sur les arbres de décision n'est donc fournie dans ce mémoire.

4.6.2.2. Régression logistique binaire

L'algorithme de désambiguïsation mis au point pour ce projet est un algorithme d'apprentissage machine de type « classification ». En effet, à chaque homographe, on doit associer une classe distincte (la « bonne » classe dans le contexte de la phrase), soit une des neuf classes grammaticales listées au Tableau 3.1. Bien qu'un code numérique ait été assigné pour chaque classe, il est à noter que la séquence des classes est arbitraire et n'a donc pas de signification. On aurait par exemple pu commencer par les interjections plutôt que par les verbes. De plus, on ne peut prétendre qu'un nom commun (code 3) est une « moyenne » ou est à mi-chemin entre un adjectif (code 2) et un adverbe (code 4). Il ne s'agit donc pas de classes « ordinales » ou même « hiérarchiques ». Ceci exclut donc d'emblée une approche axée sur la régression standard. Il est donc bel et bien question ici d'un problème de classification. C'est la raison pour laquelle on fait ici appel à la régression logistique binaire.

La régression logistique binaire permet de classifier une observation à laquelle on n'associe que deux possibilités (oui ou non, « 0 » ou « 1 », etc.), sur la base de caractéristiques quantitatives. À la base, l'algorithme de régression logistique binaire part des mêmes informations que pour les 36 arbres de décision discutés à la section précédente. Dans un cas simple où il n'existe qu'une seule caractéristique, la régression logistique binaire peut s'exprimer sous la forme d'un graphique (Figure 4.19). L'axe horizontal correspond à la valeur de la caractéristique tandis que l'axe vertical correspond à la probabilité relative des deux options (valeur entre 0 et 1).

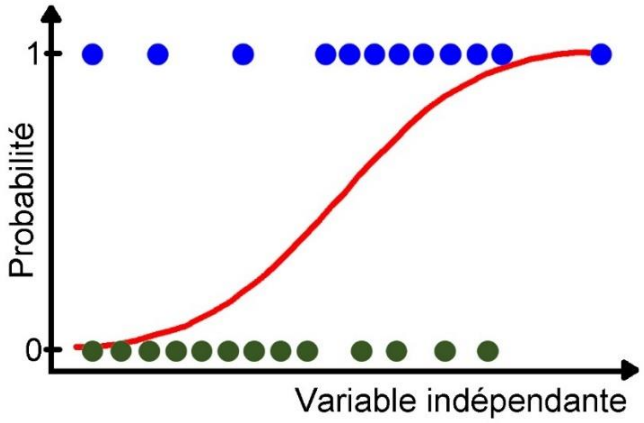


Figure 4.19 : Illustration de la régression logistique binaire dans le cas où il n'y a qu'une seule caractéristique (variable indépendante)

Le détail du fonctionnement de l'algorithme de régression logistique binaire ne sera pas discuté ici, mais un tel algorithme permet de tracer la courbe de probabilités de la Figure 4.19, en fonction de tous les points correspondant aux observations individuelles. Le cas plus général à plus de deux caractéristiques ne peut s'illustrer aussi aisément, mais le principe demeure le même : une probabilité de chacun des deux résultats possibles est calculée en fonction de la valeur de toutes les caractéristiques. L'algorithme de régression logistique associe un paramètre b_i à chacune des n caractéristiques x_i afin de minimiser les erreurs de prédiction. Ces paramètres sont appliqués dans une équation linéaire (Équation 4.3).

$$tmp = b_0 + b_1 x_1 + b_2 x_2 + (\dots) + b_n x_n \tag{4.3}$$

La variable tmp est une valeur « temporaire ». On applique ensuite une fonction dite « sigmoïde » au résultat de cette équation linéaire, pour obtenir une valeur S toujours comprise entre 0 et 1, correspondant à la probabilité d'une des deux options possibles (Équation 4.4) :

$$S(tmp) = \frac{1}{1+e^{-tmp}} \tag{4.4}$$

La valeur de probabilité S de 0.5 sert de seuil entre les deux prédictions possibles.

La régression logistique offre des avantages en comparaison avec les arbres de décision. Par exemple, toutes les caractéristiques peuvent potentiellement contribuer à la décision, grâce au paramètre associé à chacune d'elles. Ceci contraste avec les arbres de décision, qui reposent la plupart du temps sur uniquement quelques caractéristiques, à défaut de quoi l'arbre devient trop profond et complexe. Aussi, la forme de la régression logistique demeure toujours la même, soit une équation linéaire suivie de l'application de la fonction sigmoïde. Si d'autres données d'entrainement deviennent disponibles ou sont adaptées, la valeur des paramètres change, mais la structure des équations demeure identique. Encore là, cela contraste avec le cas des arbres de décision pour lesquels un changement dans les données d'entrainement peut potentiellement engendrer un changement de structure complet de l'arbre, ce qui en complique la programmation.

Le fait qu'une probabilité soit calculée pour chaque prédiction représente aussi un avantage net par rapport aux arbres de décision. D'une part, une telle information nous permet de quantifier le niveau de « certitude » de l'algorithme pour chaque cas donné. Cette information peut s'avérer pertinente si on cherche à améliorer l'algorithme en y injectant par exemple de nouvelles caractéristiques. Mais surtout, le calcul des probabilités est utile pour les cas où un homographe comporte trois formes possibles, comme le mot « ferme ». En effet, la régression logistique

binaire, par définition, ne peut considérer que deux possibilités (« binaire » = 2). Cette méthode est donc en principe inapplicable dans les cas où trois classes grammaticales sont possibles. Mais une approche alternative se basant tout de même sur les 36 régressions logistiques binaires peut être mise au point, grâce aux probabilités associées, comme on le verra à la Section 4.6.2.2.2.

4.6.2.2.1. Régression pour deux possibilités d'homographes

Tel qu'on l'a mentionné plus haut, la régression logistique binaire ne permet, par définition, de classifier des observations que selon deux catégories distinctes, pas davantage. Mais comme on l'a aussi vu à la Section 4.6.2.1, en très grande majorité, les homographes ne se rapportent qu'à deux classes grammaticales. La régression logistique binaire peut donc s'appliquer *directement* pour désambiguïser la grande majorité des homographes d'un texte.

La toute première étape pour l'entrainement de l'algorithme de régression logistique dans le cas d'homographes permettant deux classes grammaticales est de générer les 36 tableaux de caractéristiques, correspondant à toutes les paires possibles de classes (comme au Tableau 4.13). Une fois ces tableaux disponibles, on peut calculer les paramètres du modèle en utilisant un algorithme de régression logistique binaire. Le logiciel MATLAB (Mathworks, version 2021a) par exemple, contient un algorithme pré-intégré permettant la régression logistique binaire, la fonction « *mnrfit*[9] ». Et comme 36 équations de régression doivent être effectuées pour ce projet, un script MATLAB peut être créé pour générer les 36 équations de régression requises, sur la base des caractéristiques des homographes enregistrées au préalable dans des fichiers texte (extrait de code 4.5). Aucun détail supplémentaire n'est fourni ici concernant le fonctionnement de la fonction « *mnrfit* » de MATLAB.

Pour chacune des 36 paires de classes grammaticales, l'algorithme fournit en sortie autant de paramètres qu'il y a de caractéristiques dans le modèle, plus un paramètre additionnel, une constante qui représente l'ordonnée à l'origine. Ces données sont enregistrées dans un fichier texte, qui est par la suite lu par le programme Java de lemmatisation du présent projet. Le Tableau 4.14 illustre ce à quoi ressemble ce tableau de coefficients de régressions logistiques. Seulement les premières colonnes sont affichées, par souci de concision.

Ces paramètres sont ensuite appliqués aux Équations 4.3 et 4.4 pour chaque homographe rencontré dans le texte. On multiplie donc en premier lieu ces paramètres par les valeurs des caractéristiques à l'Équation 4.3, puis on applique la fonction sigmoïde de l'Équation 4.4. Si la valeur de S dans la fonction sigmoïde est supérieure à 0.5, on assigne à l'homographe la classe grammaticale correspondant au code le plus élevé parmi les deux possibles, et réciproquement. Par exemple, si on tente de départager entre un nom commun (code 3) et un adverbe (code 4) et que la valeur de S dans la fonction sigmoïde est 0.4, c'est que l'algorithme prédit que le mot en question est un nom commun.

[9] Les lettres « *mnr* » signifient « *multinomial regression* »

Extrait de code 4.5 : Script MATLAB utilisé pour générer les équations de régression logistique binaire

```
matrice=zeros(36,47);
ctr=0; % Compteur pour les 36 lignes

for i = 1:8      % Valeur de POS1
    for j= i+1:9 % Valeur de POS2

        ctr=ctr+1 ; % Compteur pour les 36 lignes
        fichier=strcat('features',int2str(i),int2str(j),'.txt');
        T = readtable(fichier);
        TA=table2array(T);
        y=TA(:,1);    % Extraire la colonne 1 pour POS
        x=TA(:,2:45); % Extraire les colonnes restantes pour facteurs et intercept
        taille=size(y,1); % Taille = nombre de lignes dans le fichier
        y=categorical(y); % Pour permettre mnrfit, y doit être categorical
        B=mnrfit(x,y);

       iVecteur=[i j]'; % Chaque ligne du fichier final commence par les deux POS
       disp(i);
       disp(j);
       B = [iVecteur; B]'; % On ajoute les deux POS aux (intercept + facteurs)
       matrice(ctr,:)=B;
    end
end
writematrix(matrice, "facteurs.txt");
```

Tableau 4.14 : Illustration du fichier en sortie des coefficients de régression logistique comprenant autant de colonnes que de caractéristiques, plus l'ordonnée à l'origine. Le fichier compte 36 lignes qui correspondent aux 36 paires de classes grammaticales distinctes. Seules quelques colonnes affichées

Code de la Classe 1	Code de la Classe 2	Coefficient 1	Coefficient 2	Coefficient 3	(..)	Ordonnée à l'origine
1	2	0.957	-0.309	-0.277		-0.0127
1	3	0.710	-0.248	-0.476		-0.0044
1	4	0.653	-0.390	-0.497		0.0308
1	5	0.257	0.078	0.012		-0.0088
1	6	0.384	-0.122	-0.232		0.0110
1	7	0.463	-0.253	-0.130		-0.0460
1	8	0.072	0.166	-0.203		-0.0099
1	9	0.596	-0.305	-0.485		0.0012
2	3	0.000	-0.044	-0.223		0.0107
2	4	0.000	-0.083	-0.212		0.0843
2	5	-0.070	0.192	0.133		0.0218
2	6	0.432	0.055	-0.023		0.0624
2	7	0.439	0.167	0.167		0.0070
2	8	0.154	0.395	0.036		0.0751
2	9	0.520	0.081	-0.365		-0.0008
3	4	0.000	-0.276	-0.083		0.0790
3	5	0.000	0.086	0.185		0.0161
3	6	0.192	0.164	0.111		0.0512
3	7	0.038	0.007	0.307		-0.0033
3	8	-0.151	0.327	0.216		0.0384
3	9	0.187	0.079	-0.009		0.0027
4	5	0.000	0.151	0.164		-0.0432
4	6	0.494	0.323	0.174		-0.0346
4	7	0.345	0.075	0.281		-0.1116
4	8	-0.301	0.377	0.136		-0.0517
4	9	0.406	0.144	0.034		-0.0004
5	6	0.403	-0.061	-0.159		0.0223
5	7	0.319	-0.054	0.073		-0.0163
5	8	0.175	-0.098	-0.203		-0.0085
5	9	0.422	-0.039	-0.094		-0.0029
6	7	0.324	-0.027	0.194		-0.0600
6	8	-0.399	0.163	0.105		-0.0083
6	9	0.338	0.018	-0.009		0.0001
7	8	-0.015	0.290	-0.114		0.0394
7	9	0.576	0.060	-0.071		0.0020
8	9	0.966	-0.095	-0.155		-0.0063

4.6.2.2.2. Régression pour plus de deux possibilités d'homographes

À la section précédente, on a *directement* appliqué un algorithme de régression logistique binaire pour les homographes ne comportant que deux possibilités de classes grammaticales. Mais bien que la grande majorité des homographes ne permettent que deux classes, il en reste environ 7%, sur la base de l'analyse des corpus de référence choisis, qui sont associés à plus de deux classes grammaticales. On y mentionnait par exemple le mot « ferme » qui offre trois possibilités (verbe, adjectif et nom) et le mot « tout » qui offre quant à lui cinq possibilités (adjectif, nom, adverbe, déterminant et pronom). Un algorithme de régression logistique binaire ne peut donc pas être appliqué directement dans de tels cas. Mais une approche basée sur la régression logistique binaire demeure tout de même possible dans les cas où plus de deux classifications sont possibles. Deux de ces approches sont décrites plus bas.

Première approche – combinaison de probabilités de paires

Une première approche consiste à identifier toutes les paires possibles de classes grammaticales pour un homographe donné. Reprenons l'exemple du mot « ferme », qui peut être soit un verbe, soit un nom ou soit un adjectif. Il y a dans ce cas trois paires possibles de classes grammaticales en considérant les trois classes associées à ce mot:

- Verbe vs. nom
- Verbe vs. adjectif
- Adjectif vs. nom

Pour cette première approche, on calcule, sur la base des valeurs des caractéristiques, un score correspondant à la classe grammaticale de verbe, en multipliant entre elles les probabilités de verbe dans les deux premiers cas (« verbe vs. nom » et « verbe vs. adjectif »). Même chose pour l'adjectif, sauf qu'on multiplie alors les probabilités d'adjectifs obtenues des deux dernières régressions (« adjectif vs. verbe » et « adjectif vs. nom »). Et finalement, on adopte la même approche pour les noms, en multipliant les probabilités que le mot soit un nom selon la première et la dernière de ces trois régressions (« nom vs. verbe » et « nom vs. adjectif »). La classe grammaticale choisie par l'algorithme correspond à celle parmi les trois affichant le score le plus élevé. Il est à noter que les trois scores obtenus ne sont pas des probabilités absolues, car leur somme est inférieure à un, dans le cas général. On parle plutôt de probabilités relatives, ce pourquoi on a ici opté pour le terme « score ». Cette même approche se généralise si un homographe appartient à quatre classes grammaticales ou plus.

Supposons par exemple qu'on calcule, pour l'homographe « ferme », les probabilités suivantes en se servant des régressions logistiques binaires :

- Verbe vs. nom 0.4 (probabilité verbe = 0.4, probabilité nom = 0.6)
- Verbe vs. adjectif 0.3 (probabilité verbe = 0.3, probabilité adjectif = 0.7)
- Adjectif vs. nom 0.8 (probabilité adjectif = 0.8, probabilité nom = 0.2)

Nous obtenons alors les trois scores suivants :

- Score verbe = 0.4 x 0.3 = 0.12
- Score nom = 0.6 x 0.2 = 0.12
- Score adjectif = 0.7 x 0.8 = 0.56

Comme le score le plus élevé est 0.56, l'algorithme en déduit que le mot a davantage de chances d'être un adjectif. L'algorithme de désambiguïsation fournit donc en sortie pour le mot « ferme » dans ce contexte, la classe grammaticale « adjectif ». Et on peut ensuite estimer la probabilité que cette prédiction soit correcte, en fonction des scores plus haut, en effectuant le calcul présenté à l'Équation 4.5 :

$$P_{\text{adjectif}} = \frac{S_{\text{adjectif}}}{(S_{\text{verbe}}+S_{\text{nom}}+S_{\text{adjectif}})} = \frac{0.56}{0.12+0.12+0.56} = 0.7 \quad (4.5)$$

Où P est la probabilité et S est le score, tel que calculé plus haut. On obtient donc une probabilité de 0.7, ce qui équivaut à 70%. On voit donc que suivant cette approche, on peut utiliser les régressions logistiques binaires, même dans les cas d'homographes impliquant plus de seulement deux possibilités. On peut donc faire des prédictions pour tous les cas d'homographes, en n'utilisant que les 36 équations de régression logistique correspondant aux tableaux n'impliquant que des paires de classes grammaticales.

La Figure 4.20 illustre le cas de l'homographe « tout » qui comporte cinq possibilités de classes grammaticales. Les probabilités issues des régressions logistiques binaires (Équations 4.3 et 4.4) sont incluses dans les différentes cases du tableau de la Figure 4.20. Par exemple, on obtient une probabilité de 0.061 pour le cas « adjectif vs. nom commun » (ovale bleu). On en déduit que dans le contexte de la phrase sous étude, le mot « tout » a bien plus de chances d'être un adjectif qu'un nom commun. L'ovale rouge quant à lui, correspondant au cas « nom commun » vs. « adjectif » indique une probabilité de 0.939, qui est obtenue tout simplement par la soustraction suivante :

$$1 - 0.061 = 0.939 \quad (4.6)$$

En effet, dans le cas d'une régression logistique binaire, la somme des deux possibilités est toujours égale à un. Les éléments du Tableau 4.20 apparaissant au bas de la diagonale peuvent donc être calculés en effectuant des soustractions comme celle de l'Équation 4.6. Un score est ensuite calculé pour chaque classe, en multipliant tous les éléments de la ligne correspondante. Le score le plus élevé détermine la classe grammaticale la plus probable.

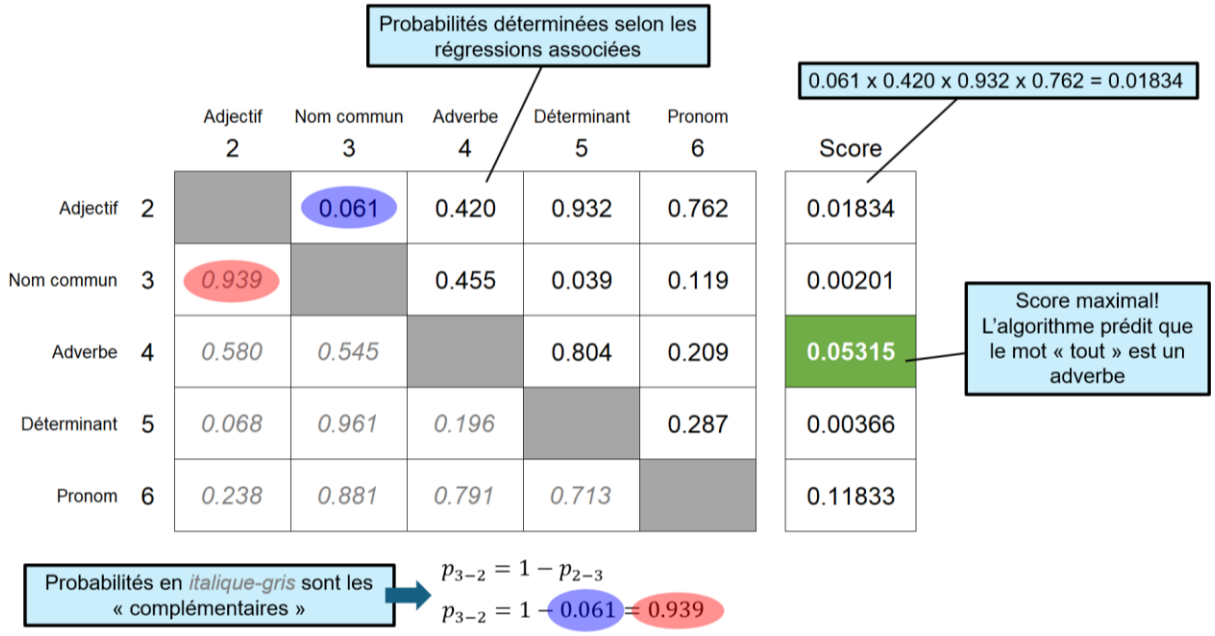


Figure 4.20 : Illustration de l'approche par combinaison de probabilités de paires de classes grammaticales pour la désambiguïsation d'homographes contenant plus de deux classes (ici, le mot « tout » en contient 5)

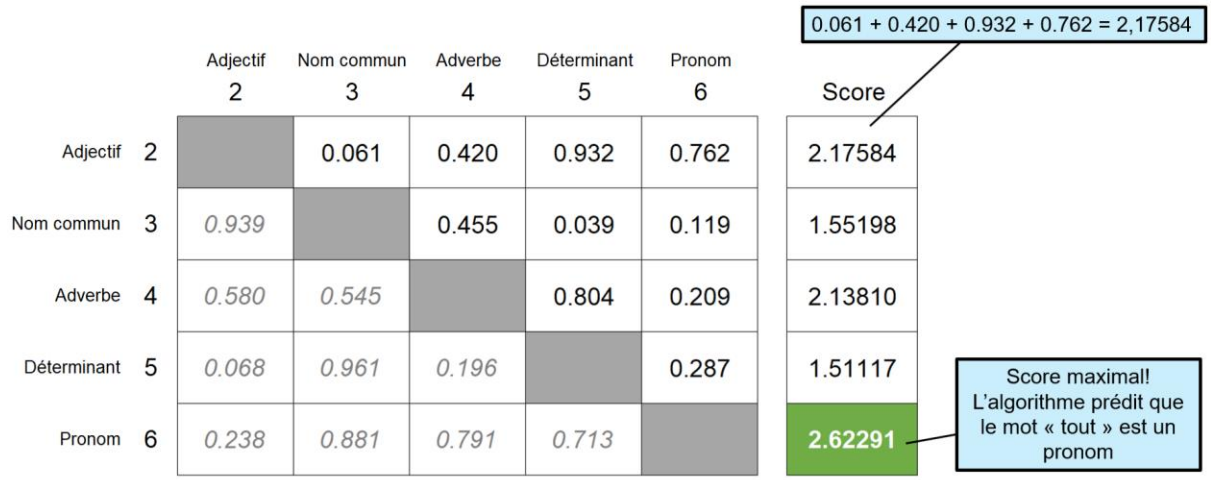


Figure 4.21 : Illustration de l'approche par combinaison de probabilités de paires de classes grammaticales pour la désambiguïsation d'homographes contenant plus de deux classes. Mais en comparaison avec la Figure 4.20, le score est ici obtenu par l'addition des probabilités, plutôt que leur produit

Il est toutefois à noter qu'on aurait pu opter, pour le calcul du score, pour la *somme* des probabilités, plutôt que pour leur *produit*, comme on l'a fait à la Figure 4.20. À la Figure 4.21, basé sur le même exemple du mot « tout », on calcule en effet le score pour chaque classe grammaticale en additionnant les éléments de chaque ligne. Tout comme on l'a fait précédemment, on détermine la classe la plus probable en repérant le score le plus élevé. On remarque que la prédiction obtenue sur la base de l'addition (« pronom », Figure 4.21) est différente que celle obtenue sur la base du produit (« adverbe », Figure 4.22). Mais pour ce projet, l'approche basée sur la multiplication (Figure 4.21) a été choisie, puisqu'elle a mené à une meilleure performance de désambiguïsation.

Deuxième approche – une classe contre toutes

La deuxième approche pour la désambiguïsation d'homographes offrant plus de deux possibilités de classes grammaticales consiste à calculer les probabilités qu'un mot appartienne à chacune des classes permises, en comparaison avec les autres. Par exemple, pour le mot « ferme », il y a trois possibilités : verbe (code=1), adjectif (code=2) et nom commun (code=3). On calcule donc pour ce mot trois probabilités :

- Probabilité que le mot soit un verbe vs. pas un verbe
- Probabilité que le mot soit un adjectif vs. pas un adjectif
- Probabilité que le mot soit un nom commun vs. pas un nom commun

Pour calculer la première régression (verbe vs. pas un verbe), on conserve le code « 1 » pour toutes les observations du corpus où le mot « ferme » est un verbe. Cependant, pour toutes les observations où le mot « ferme » est soit un adjectif (code « 2 ») soit un nom commun (code « 3 »), on remplace ces codes par la valeur « 0 ». Ainsi, on se retrouve avec deux possibilités uniquement, c'est-à-dire que soit le mot est un verbe, soit il n'en est pas un. Le fait qu'on ne retrouve que deux possibilités nous permet de *directement* utiliser la régression logistique binaire pour calculer la probabilité.

On effectue ensuite les mêmes opérations pour évaluer la probabilité que le mot « ferme » soit un adjectif dans le contexte en jeu. Cette fois, on conserve le code « 2 » pour toutes les observations du corpus où le mot « ferme » est un adjectif. Et on assigne le code « 0 » à toutes les observations où le mot « ferme » est soit un verbe (code « 1 ») soit un nom commun (code « 3 »). On peut donc

une fois de plus appliquer une régression logistique binaire, puisqu'on fait face de nouveau à seulement deux options. Le même principe s'applique au calcul de la probabilité que le mot « ferme » soit un nom commun.

La Figure 4.22 illustre le procédé correspondant à cette deuxième approche, avec un exemple fictif correspondant à un homographe de type « verbe-adjectif-nom ». Le tableau avec trois possibilités de classes grammaticales (à la gauche) est remplacé à sa droite par trois tableaux « simplifiés » correspondant chacun à une des trois classes en particulier.

On constate donc qu'il nous faut ici calculer trois nouvelles régressions logistiques binaires, chacune correspondant à un des trois tableaux à la droite de la Figure 4.22. C'est là un désavantage de cette deuxième approche, puisqu'on se rappellera que dans la première approche, on utilisait les mêmes 36 équations de régressions déjà calculées pour les cas où seulement deux classes sont possibles. Il suffisait de les combiner judicieusement.

La première approche est finalement celle qui a été choisie pour l'algorithme général de désambiguïsation, en partie justement par le fait qu'elle ne requiert pas de calculs de régression supplémentaires. Aussi, sa performance s'est montrée supérieure à la deuxième approche, dans le cas général. On verra cependant, à la Section 4.6.5, que la deuxième approche s'est tout de même avérée utile pour certains tests de désambiguïsation spécialisés.

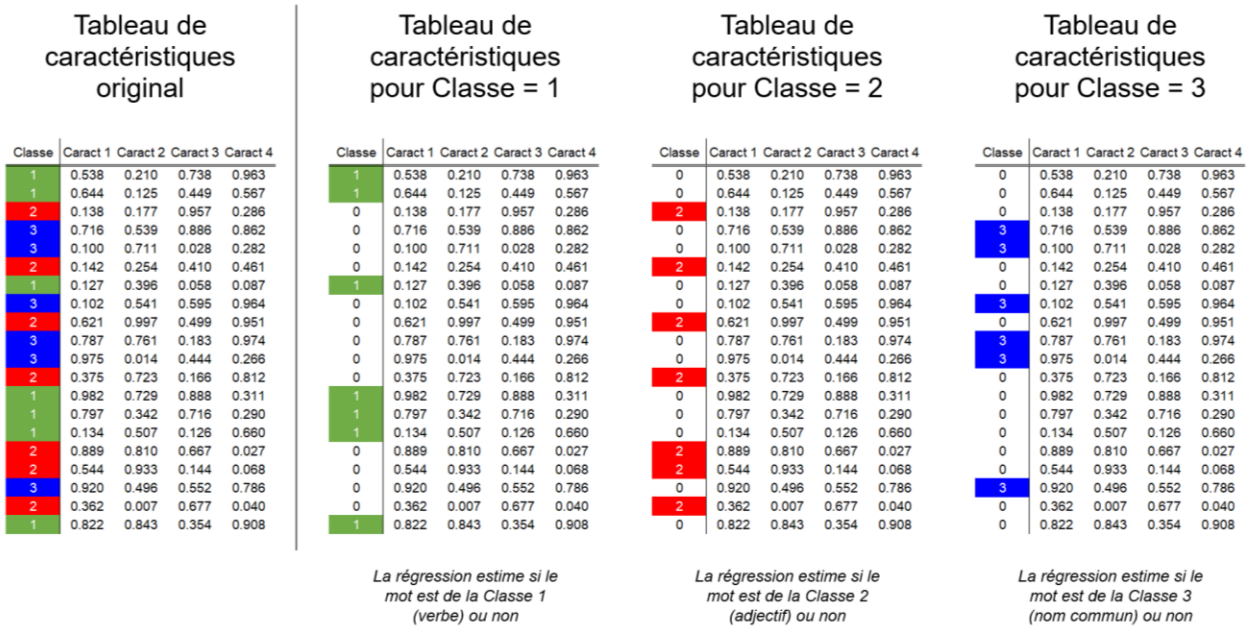


Figure 4.22 : Illustration (exemple fictif) de l'approche visant à calculer la probabilité de chacune des classes grammaticales pour la désambiguïsation d'homographes de type « verbe-adjectif-nom ». Dans chaque cas, on réduit l'analyse à deux cas : l'homographe fait partie de la classe en question, ou non

4.6.3. Entraînement du modèle

Pour ce projet, les observations qui nous intéressent concernent les homographes, leurs classes grammaticales et leurs caractéristiques associées. L'entrainement consisterait donc normalement à étiqueter manuellement un à un, bon nombre d'homographes, c'est-à-dire d'en déterminer la classe grammaticale dans le contexte des phrases du corpus. En parallèle, on évaluerait leurs caractéristiques (Section 4.6.1). Puis, à partir de ces données, l'algorithme développerait les paramètres pour en optimiser la classification. Cette étape d'entrainement comporte donc généralement une opération manuelle, car un humain doit lui-même procéder à la classification pour que l'algorithme *apprenne* à partir du savoir de l'humain. Plus l'algorithme dispose d'observations étiquetées pour l'entrainement, plus le modèle est efficace pour la prédiction.

Hélas, cet étiquetage manuel est long et laborieux pour un long texte. De nombreux chercheurs ont développé des outils de désambiguïsation basés sur l'apprentissage machine, en se basant sur de larges corpus déjà manuellement lemmatisés (Bourdaillet & Ganascia, 2005). Lors de l'utilisation de tels corpus déjà lemmatisés, l'étape d'entraînement est donc déjà partiellement effectuée, ce qui épargne donc les chercheurs faisant appel à ces corpus un temps très précieux.

Mais pour le travail actuel, on se rappelle qu'on s'intéresse à un corpus de référence précis, à partir duquel on extrait le lexique requis pour la génération de phrases au hasard. C'est donc à partir de ce lexique qu'on veut déterminer les paramètres de tout modèle basé sur l'apprentissage machine. Il faut donc effectuer l'entraînement de l'apprentissage machine sur la base de ce corpus de référence en particulier. On ne peut donc pas compter sur le travail d'entraînement fastidieux déjà effectué par d'autres chercheurs.

Mais tel qu'on l'a mentionné précédemment, la désambiguïsation des homographes n'est pas le but premier de ce projet qui est d'abord et avant tout de générer des phrases aléatoires automatiquement lemmatisées dont on se servira par la suite pour l'évaluation d'outils de lemmatisation existants. La désambiguïsation ici ne sert qu'à assurer une plus grande ressemblance entre le lexique du corpus de référence et celui généré dans les phrases aléatoires. Il faut donc considérer qu'une haute performance de l'outil de désambiguïsation, bien que souhaitable et avantageuse, n'est pas pour autant critique dans le contexte de ce projet.

Nous allons donc en premier lieu proposer une approche d'apprentissage machine simple limitant le plus possible toute intervention humaine (manuelle) lors de l'entrainement. En effet, tel que mentionné plus haut, les outils de lemmatisation existants font souvent appel à de grands corpus lemmatisés manuellement à grands efforts, pour l'entrainement et la validation de leurs outils de désambiguïsation. Cette étape manuelle ne sera pas nécessaire ici. Cette approche dite « automatique », bien que limitant la performance attendue de l'algorithme, fera gagner un temps précieux en automatisant complètement le processus. L'approche automatique consiste à n'utiliser que les mots du corpus qui ne sont *pas* des homographes. La classe grammaticale de ces non-homographes est forcément connue, dans la mesure évidemment où ces mots ont été inclus aux banques de mots de départ (table de hachage « *tableRef* »). Ainsi, on peut assigner automatiquement leur classe grammaticale lors de la création des fichiers texte de caractéristiques (Tableaux 4.11 et 4.13), et y incorporer les caractéristiques correspondant à ces mots. Lorsqu'on fait ainsi appel à tous les « non-homographes », on qualifie l'approche automatique de *complète*. Cependant, dans le but d'augmenter la performance de désambiguïsation, une autre approche a finalement été adoptée pour ce projet. Cette approche consiste à ne considérer que les « non-homographes » non adjacents à des homographes. Autrement dit, on ne considère que les « non-homographes » étant soit entourés de deux « non-homographes », ou soit en début ou en fin de phrase, dans la mesure où ils ne sont pas adjacents à un homographe. On verra plus loin (Section 4.6.3.1) l'avantage d'une telle approche plus restrictive dans la sélection des mots à inclure pour l'entrainement.

Afin d'atteindre une meilleure performance de désambiguïsation et de fournir une comparaison avec l'approche automatique décrite plus haut, l'approche traditionnelle d'entrainement impliquant une lemmatisation manuelle sera elle aussi utilisée. Mais considérant l'effort considérable requis pour cette approche manuelle, seule une partie restreinte des corpus de référence sera lemmatisée manuellement pour l'entrainement manuel de l'algorithme. On verra au Chapitre 5 que même une lemmatisation manuelle partielle du corpus de référence peut mener à d'excellents résultats pour la désambiguïsation. Pour l'approche manuelle, on peut inclure *tous* les mots du corpus, qu'ils soient homographes ou non.

La Figure 4.23 compare le fonctionnement de l'entrainement, dans le cas de l'analyse automatique et dans le cas de l'analyse manuelle, telles que définies plus haut. En utilisant une phrase extraite du roman « Le Rouge et le Noir », on comptabilise les mots qui serviront à l'entrainement pour ces deux approches. Tel qu'on l'a mentionné plus haut, l'analyse automatique peut être effectuée de façon « complète », c'est-à-dire en considérant tous les « non-homographes » du corpus. Elle peut aussi être effectuée de façon « limitée », en ne considérant que les « non-homographes » non adjacents à des homographes. Ces deux options sont aussi illustrées à la Figure 4.23.

Ces deux grandes approches pour l'entrainement (« automatique » et « manuelle ») sont le sujet des deux prochaines sections.

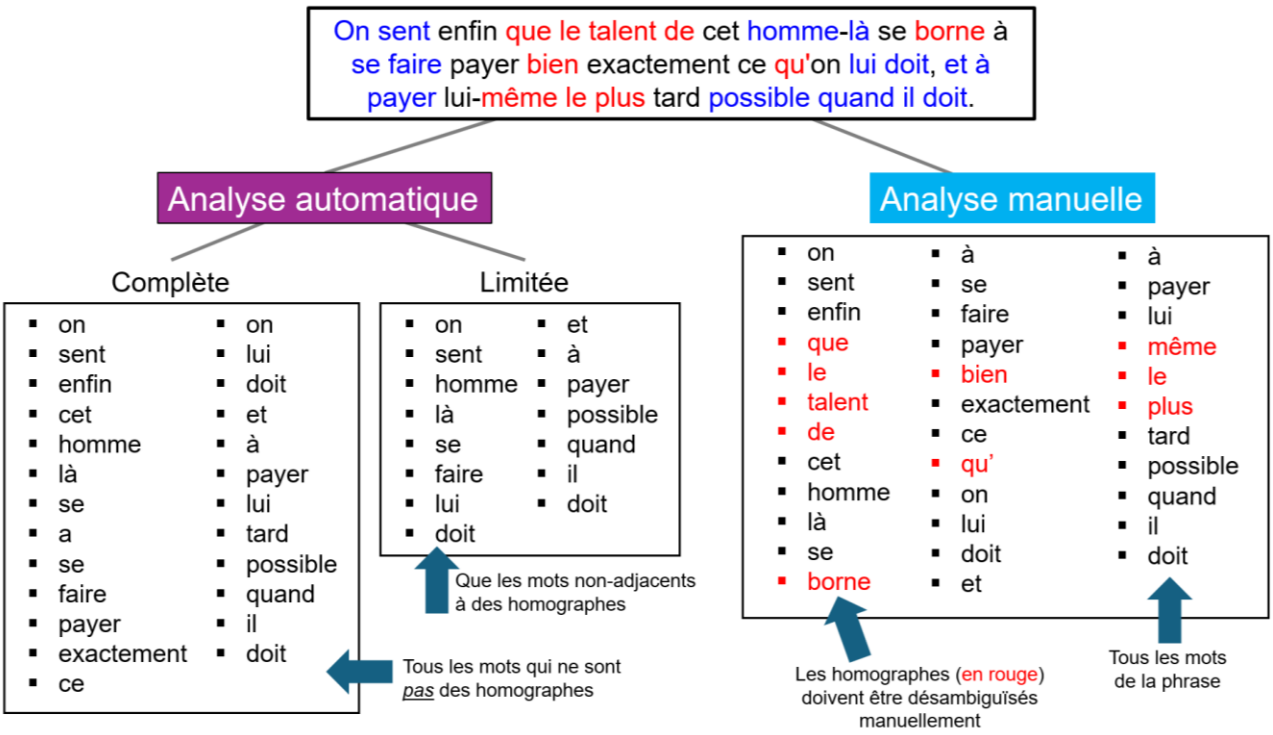


Figure 4.23 : Liste des mots inclus pour les trois types d'entrainement, dans le cas d'une phrase extraite du roman « Le Rouge et le Noir » comportant plusieurs homographes (en rouge). En bleu, on retrouve les « non-homographes » utilisés dans la version limitée de l'approche automatique

4.6.3.1. Entraînement automatique

Pour l'entraînement automatique, on s'éloigne de l'approche traditionnelle et manuelle de désambiguïsation des homographes, en tirant profit du fait que la majorité des mots du corpus ne sont pas des homographes[10]. Pour ce type d'entrainement, on choisit ainsi d'ignorer les homographes complètement, et donc de se concentrer sur les « non homographes » ou mots « non ambigus ». Autrement dit, les mots dont on connaît la classe grammaticale avec certitude, puisqu'il n'en existe qu'une seule possible. Prenons par exemple la phrase suivante :

« Cette chaise serait davantage confortable sans accoudoirs. »

Tous les mots de cette phrase ont été choisis sur la base qu'ils sont non ambigus : aucun d'eux n'est un homographe. Il en résulte que la classe grammaticale de chacun est forcément unique et connue. Donc, l'étiquetage de tous les mots de cette phrase ne requiert *aucune* intervention humaine puisque tous ne contiennent qu'une seule valeur de la variable « POS » à la table de hachage « *tableRef* ». On peut donc « nourrir » l'algorithme avec des observations issues de tels mots non ambigus, en y apposant leurs caractéristiques associées. Cette approche automatique semble donc très avantageuse.

Une ombre au tableau : les caractéristiques telles que celles décrites aux Tableaux 3.13 et 3.14 requièrent en principe que les classes grammaticales de la plupart des autres mots de la phrase soient connues. En effet, certaines caractéristiques sont basées sur la classe du mot précédent, ou sur celle du suivant. D'autres sont basées sur l'accord des déterminants, noms et adjectifs, ce qui demande de pouvoir classifier ces mots. Certaines caractéristiques sont plutôt basées sur la conjugaison des verbes, demandant évidemment qu'on puisse identifier les formes verbales et les mots environnants ayant une influence sur elles. Mais alors que la phrase citée plus haut ne contient aucun homographe, de telles phrases sont dans les faits rarissimes. En effet, des mots comme « le », « la », « les », « l ' », et « de » sont des homographes et se retrouvent dans presque toutes les phrases. La très grande majorité des phrases contient donc au moins un homographe, ce qui complique le « calcul » des caractéristiques.

La façon de contourner ce problème qui a été adoptée pour ce projet est la suivante. Pour chaque homographe autre que le mot sous analyse, on considère en simultané toutes ses variantes possibles pour déterminer les valeurs des caractéristiques. Cette approche fausse forcément les résultats, puisque dans un contexte précis, un mot ne peut pas avoir deux fonctions grammaticales simultanément. Mais cette faiblesse est acceptée pour l'approche actuelle, reconnaissant que l'ambigüité qui en résulte est compensée par l'avantage marqué de ne pas avoir à lemmatiser le texte manuellement.

Comme on se base sur les mots non ambigus, l'entrainement peut donc se faire de façon automatique avec un très grand nombre de mots. Mais pour tout de même diminuer l'impact négatif de l'ambiguïté associée aux homographes, sans pour autant l'éliminer, on a opté pour ce projet d'accepter à l'étape d'entrainement uniquement certains mots non ambigus. Tel que mentionné plus haut, on n'accepte à l'entrainement que les mots non ambigus non adjacents à des homographes. Autrement dit, on accepte les mots non ambigus qui sont précédés d'un autre mot non ambigu et qui sont suivis d'un autre mot non ambigu. On considère aussi le premier mot de la phrase, si non ambigu et suivi d'un non ambigu, et le dernier mot de la phrase, si non ambigu et précédé d'un autre non ambigu. Même après l'application de ce « filtre », il reste tout de même plusieurs dizaines de milliers d'observations aux corpus de référence choisis, pouvant alimenter l'algorithme lors de l'entrainement. La Figure 4.23 donne des exemples de mots non ambigus (en bleu dans la phrase) pouvant servir à l'entrainement, selon cette règle. En effet, la classe

[10] On verra au Chapitre 5 que les homographes représentent environ le tiers de tous les mots du corpus de référence

grammaticale des mots non ambigus de la Figure 4.23 est toujours connue, car unique et déjà incorporée à la table de hachage « *tableRef* ».

Comme on le rappelait plus haut, ces caractéristiques s'évaluent au niveau de chaque phrase prise indépendamment. Toutes les phrases, dont le début et la fin ont précédemment été identifiés grâce à la présence de points (voir Section 4.2.3), sont donc envoyées l'une après l'autre à l'algorithme servant à calculer les caractéristiques. Mais la présence d'homographes dans les phrases complique l'évaluation de ces caractéristiques. Prenons la phrase suivante en exemple :

« L'officier montre la ferme rose de son commandant ».

Dans cette phrase, tous les mots sont des homographes. A priori, on ne peut donc pas déterminer si :

- « l' » est un pronom ou un déterminant
- « officier » est un nom commun ou un verbe
- « montre » est un nom commun ou un verbe
- « la » est un déterminant ou un pronom
- « ferme » est un nom commun, un adjectif ou un verbe
- « rose » est un adjectif ou un nom commun
- « de » est un déterminant ou une préposition
- « son » est un nom commun ou un déterminant
- « commandant » est un nom commun ou un verbe

Devant toute cette incertitude, il devient difficile d'appliquer les règles de grammaire des Tableaux 3.13 et 3.14 du Chapitre 3, puisque ces règles supposent que l'on connaît les classes grammaticales de tous les mots pertinents de la phrase. En cas d'homographes, l'algorithme développé en considère donc toutes les possibilités, selon l'approche adoptée ici.

Par exemple, le mot « ferme » sera considéré comme un verbe conjugué pour vérifier la règle comme quoi il devrait y avoir au moins un verbe conjugué dans la phrase, en assumant donc que ce mot est en effet un verbe conjugué. Mais on vérifiera aussi si le mot « ferme » est précédé d'un déterminant, en supposant cette fois que le mot « ferme » est un nom commun. Les Tableaux 4.15a et 4.15b présentent les « valeurs » des caractéristiques des Tableaux 3.13 et 3.14 pour le cas du mot « ferme » apparaissant à la phrase citée plus haut.

Dans l'analyse du Tableau 4.15a pour le mot « ferme » dans la phrase choisie, on constate qu'une des caractéristiques a pris la valeur « oui » (1), de façon erronée, car cette valeur était basée sur l'hypothèse que le mot « ferme » puisse ici être un verbe, ce qui n'est pas le cas. Mais comme l'algorithme évalue ces caractéristiques *avant* de déterminer les classes des homographes, il ne peut exclure cette possibilité. Il en découle que beaucoup d'homographes, au moment de l'évaluation des caractéristiques, seront associés à des combinaisons de caractéristiques qu'on ne retrouverait pas si on en effectuait plutôt l'analyse manuelle détaillée. C'est là une faiblesse de cette approche « automatique », qui mène possiblement à une performance moindre qu'une analyse manuelle détaillée. Au Chapitre 5, on pourra comparer la performance de désambiguïsation basée sur l'entrainement automatique ou manuel, pour s'en convaincre.

Tableau 4.15a : Évaluation des caractéristiques du mot « ferme » dans la phrase « L'officier montre la ferme rose de son commandant. ». Caractéristiques en lien avec les verbes

Règle de grammaire	Résultat
Si précédé d'un verbe attributif, le nombre doit concorder	Non (0). Aucun mot précédant le mot « ferme » n'est un verbe attributif.
Généralement précédé d'un auxiliaire « être » ou « avoir », ou d'un verbe attributif. Les deux mots peuvent être séparés par un adverbe.	Non (0). Le mot « ferme » n'est pas précédé plus tôt dans la phrase d'un auxiliaire « être » ou « avoir ».
On retrouve souvent un participe passé suite à un auxiliaire « être » ou « avoir ». Le participe passé n'est pas toujours situé directement après l'auxiliaire. Note : les verbes « être » ou « avoir » ne sont pas uniquement utilisés comme auxiliaires.	Non (0). Le mot « ferme » n'est pas un auxiliaire.
On ne retrouve pas deux verbes conjugués qui se suivent	Non (0). Le mot qui suit le mot « ferme » est « rose ». Ce mot n'est pas un verbe conjugué.
Si un pronom personnel sujet (nominatif) précède un verbe, le verbe doit se conjuguer à la même personne que le pronom personnel en question. Note : La règle s'applique aussi pour les groupes nominaux employés comme sujets « les animaux mangent », mais cette règle ne sera pas appliquée pour ce projet, pour limiter la complexité des algorithmes.	Non (0). Le mot « ferme » est possiblement un verbe, mais le mot qui le précède (« la ») n'est pas un pronom sujet (c'est plutôt possiblement un pronom objet).
Une phrase contient généralement au moins un verbe conjugué.	(2) On identifie la présence de deux verbes conjugués potentiels dans la phrase (« montre » et « ferme »).
Dans les phrases négatives, on retrouve généralement un verbe conjugué entre les adverbes « ne » et « pas ». Il peut y avoir d'autres mots entre ces trois. D'autres adverbes jouent parfois le même rôle que « pas » : « plus », « point », « guère »	Non (0). On ne retrouve ni le mot « ne » ni le mot « pas » (ou équivalent) de part et d'autre du mot « ferme »
À la suite des prépositions « à », « de », « pour » et « sans », un verbe doit se retrouver à la forme infinitive.	Non (0). Le mot précédant le mot « ferme » est « la ». Ce mot n'est pas une préposition
Dans une phrase négative, c'est une forme infinitive qui suit normalement les mots « ne » et « pas »	Non (0). On ne retrouve pas les prépositions « ne » et « pas » devant le mot « ferme ».

Tableau 4.15b : Évaluation des caractéristiques du mot « ferme » dans la phrase « L'officier montre la ferme rose de son commandant. ». Caractéristiques non reliées aux verbes

Règle de grammaire	Résultat
Si précédé d'un déterminant, le mot doit s'accorder en genre et en nombre avec le déterminant.	Oui (1). Le mot « ferme » est possiblement un nom, précédé du mot « la », qui est possiblement un déterminant. Ces deux mots s'accordent en genre (féminin) et en nombre (singulier)
Si suivi d'un adjectif postposé, le nom doit s'accorder en genre et en nombre avec l'adjectif. Permettre la présence d'un adverbe entre les deux.	Oui (1). Le mot qui suit le mot « ferme » est « rose ». Ce mot est possiblement un adjectif postposé. Ces deux mots s'accordent en genre (féminin) et en nombre (singulier)
Si précédé d'un adjectif antéposé, le nom doit s'accorder en genre et en nombre avec l'adjectif.	Non (0). Le mot précédant le mot « ferme » est « la ». Ce mot n'est pas un adjectif antéposé.
S'accorde en genre et en nombre avec le nom commun qui le suit.	Oui (1). Le mot qui suit le mot « ferme » est « rose ». Ce mot est possiblement un nom commun, et s'accorde en genre et en nombre avec « ferme », en supposant que celui-ci ne soit pas un verbe. Une analyse manuelle nous dirait que le mot « rose » ici n'est pas un nom commun, ce qui devrait donner la réponse « Non (0) », mais du point de vue de l'algorithme, la classe grammaticale de « rose » reste incertaine.
S'accorde en genre et en nombre avec le nom commun qui le précède. Permettre la présence d'un adverbe entre les deux.	Non (0). Le mot précédant le mot « ferme » est « la ». Ce mot n'est pas un nom commun.
S'accorde en genre et en nombre avec le nom commun, l'adjectif ou le déterminant numéral qui le suit.	Oui (1). Le mot qui suit le mot « ferme » est « rose ». Ce mot est possiblement un adjectif. Ces deux mots s'accordent en genre (féminin) et en nombre (singulier)

4.6.3.2. Entraînement manuel

L'entrainement manuel est le type d'entrainement traditionnel pour l'apprentissage machine. Il consiste à classifier manuellement chaque observation considérée lors de l'entrainement. Dans le cas qui nous intéresse, on assigne donc manuellement une des 9 classes grammaticales à chacun des homographes que l'on rencontre. Heureusement, comme on le mentionnait plus haut, seule une minorité des mots des textes en français sont des homographes (environ 30%), si bien que pour le reste (70%), aucun effort manuel n'est requis. En effet, l'algorithme de lemmatisation de base (Section 4.3) arrive à identifier avec certitude la classe des non homographes selon la seule classe apparaissant pour eux dans la table de hachage « *tableRef* ». L'utilisation des non homographes pour l'entrainement procure donc un fort effet de levier, puisqu'en désambiguïsant manuellement 3 mots, on en retire en moyenne 10 pour l'entrainement.

Pour faciliter l'entrainement manuel, un algorithme Java a été mis au point pour repérer chaque homographe et fournir la phrase complète où il apparait. La Figure 4.24 donne l'exemple de la sortie de cet algorithme pour le premier homographe du roman « Le Rouge et le Noir », apparaissant à la première phrase du roman. Il suffit alors, une fois qu'une classe grammaticale est assignée à chaque homographe, de comptabiliser le tout dans un fichier texte. Le Tableau 4.16 fournit les quelques premières lignes de ce fichier, pour le roman « Le Rouge et le Noir ». On y retrouve en effet les classes grammaticales de tous les homographes de la première

phrase apparaissant aussi à la Figure 4.24. Par exemple, le premier homographe (« la ») est classifié comme un déterminant (code 5), le deuxième homographe (« de ») est classifié comme une préposition (code 7), le troisième homographe (« l' ») est classifié comme un déterminant (code 5), et ainsi de suite. Ce fichier texte sera ensuite lu au moment de l'exécution de l'algorithme d'entrainement.

Tel qu'on l'a mentionné plus haut, la désambiguïsation manuelle est fastidieuse, si bien que seule une partie de chaque corpus de référence a été désambiguïsée manuellement, correspondant tout de même à plus de 10 000 homographes dans chaque cas » Pour ces 10 000 homographes, on se retrouve donc avec un total de plus de 30 000 mots disponibles pour l'entrainement, en incorporant aussi les mots qui ne sont pas des homographes.

L'efficacité de l'entrainement dépend évidemment, en plus de la quantité d'observations disponibles, de la qualité du travail de désambiguïsation. Toute opération humaine de ce genre injecte un élément de risque, puisque des erreurs peuvent se glisser. Par exemple, dans une phrase donnée, l'homographe « de » pourrait avoir été manuellement classifié comme un déterminant, alors qu'il est en réalité une préposition dans le contexte de la phrase. Si de telles erreurs sont peu nombreuses, l'efficacité de l'algorithme ne devrait pas trop en souffrir. Heureusement, il peut arriver que l'algorithme lui-même mette en lumière de telles erreurs de désambiguïsation manuelle, au moment de l'évaluation de la performance de l'algorithme. Cette évaluation du modèle est le sujet de la prochaine section.

```
---------------------------------------------------------------

1 Homographe! Mot =la

Phrase:  LA petite ville de Verrières peut passer pour l' une des
plus jolies de la franche Comté

---------------------------------------------------------------
```

Figure 4.24 : Sortie de l'algorithme Java donnant l'information de base facilitant la désambiguïsation manuelle des homographes. L'homographe est en majuscules dans la phrase, ce qui permet de bien le repérer, ce qui est très utile quand un même homographe apparait plus d'une fois dans la même phrase, comme c'est le cas ici (le mot « la » est utilisé deux fois dans cette phrase)

Tableau 4.16 : Classes grammaticales des premiers homographes du roman « Le Rouge et le Noir » déterminées manuellement. Information enregistrée dans un fichier texte (sans l'entête)

Homographe	Classe grammaticale
la	5
de	7
l'	5
plus	4
de	7
la	5
leurs	5

4.6.4. Évaluation et application du modèle

Une fois l'entrainement effectué, que ce soit selon l'approche automatique (Section 4.6.3.1) ou selon l'approche manuelle (Section 4.6.3.2), tous les paramètres du modèle d'apprentissage machine sont en place pour le mettre en application. De façon générale, on applique le modèle d'apprentissage machine dans deux contextes : l'évaluation du modèle et l'application du modèle à de nouvelles données inconnues. L'évaluation consiste à appliquer le modèle sur un ensemble de données dont on connaît la classification. Dans le contexte de ce projet, cela correspond à appliquer le modèle à des homographes dont on a déjà manuellement déterminé la classe grammaticale. Comme on connaît d'avance la classe, il est alors possible de comparer la prédiction du modèle avec la classe grammaticale réelle. L'outil typique utilisé pour quantifier la performance de modèles de classification est la matrice de confusion, dont on discutera à la Section 4.6.4.2 dans le contexte spécifique de ce projet. Mais tandis que l'étape d'évaluation ne sert qu'à valider le modèle, l'application du modèle quant à elle, est le but ultime d'un outil de lemmatisation pour des applications pratiques, telles que discutées au Chapitre 1 : la quantification de la richesse lexicale, l'assistance à la traduction automatique, etc.

Mais avant de procéder à l'évaluation ou à l'application, on doit déterminer quels jeux de données serviront à l'étape d'entrainement, puis à l'étape d'évaluation, le cas échéant. Ces jeux de données sont discutés à la section suivante.

4.6.4.1. Ensemble d'entrainement et ensemble de test

Lorsqu'il est question d'entrainement et d'évaluation de modèles d'apprentissage machine, on définit deux jeux de données : l'ensemble d'entrainement, et l'ensemble de test. L'ensemble d'entrainement, comme son nom l'indique, contient l'ensemble des données qui servent à entrainer le modèle. L'ensemble de test quant à lui, contient l'ensemble des données servant à comparer le résultat des prédictions du modèle avec les valeurs de classifications connues.

Il est tentant de tester et d'évaluer le modèle sur la base des mêmes données. Dans un tel cas, l'ensemble de test est le même que l'ensemble d'entrainement (Figure 4.25a). Dans un tel cas, on utilise l'entièreté des données pour l'entrainement du modèle, ce qui est avantageux, car plus on dispose de données, plus le modèle a le potentiel d'être robuste. De plus, comme on utilise aussi l'entièreté des données pour l'évaluation, on peut quantifier la performance du modèle sur la base d'un échantillon de taille maximale. Cependant, il n'est jamais recommandé d'utiliser le même jeu de données pour l'entrainement et le test, car on introduit alors un risque de surapprentissage. Le surapprentissage consiste à spécialiser un modèle pour un jeu de données précis, au détriment de sa performance dans des situations plus générales. De plus, le niveau de performance obtenu est alors trompeur, car on évalue l'efficacité du modèle à même les données utilisées pour l'entrainement.

Il est plutôt recommandé d'évaluer l'efficacité d'un algorithme de classification en utilisant un jeu de données de test complètement indépendant du jeu de données d'entrainement. Bien que ce choix soit jusqu'à un certain point arbitraire, il est souvent recommandé d'utiliser 70% des données pour l'entrainement, et le 30% restant pour l'évaluation (Figure 4.25b).

Mais pour le projet actuel, dans le cas de l'entrainement automatique (Section 4.6.3.1), la situation est différente. On se rappelle que pour effectuer l'entrainement automatique, on n'utilise que les mots non ambigus (non homographes), tandis que pour l'évaluation, on n'utilise que les homographes. Il n'y a donc aucun chevauchement possible dans ce cas entre le jeu de données d'entrainement et le jeu de données de test, puisqu'aucun mot ne peut être à la fois un homographe et un non homographe (Figure 4.25c). Il paraît donc plus acceptable d'utiliser le même texte pour l'entrainement et l'évaluation dans le cas de l'approche automatique. Mais

encore là, on pourrait arguer que bien que les mots des deux ensembles ne soient pas les mêmes, ils sont issus des mêmes phrases, donc pas complètement indépendants.

Une autre approche pour l'évaluation d'algorithmes de classification est la méthode « *k-fold* ». Celle-ci consiste à séparer le jeu complet d'observation en un nombre *k* de sous-ensembles, ou la valeur *k*=10 est souvent celle employée. À tour de rôle, chaque sous-ensemble sert alors comme jeu de test, alors que les 9 autres sous-ensembles servent à l'entrainement. On se retrouve donc à effectuer 10 itérations (Figure 4.26). On calcule finalement la performance du modèle en calculant la moyenne de sa performance sur les 10 itérations. C'est une des approches qui sera adoptée pour ce projet, lors de la présentation des résultats au Chapitre 5.

Finalement, comme on le mentionnait à la Section 3.4, deux corpus de références distincts ont été sélectionnés pour ce projet (un roman français du 19e siècle, et un roman de science-fiction contemporain). Une autre approche pour l'évaluation de l'algorithme de désambiguïsation sera d'utiliser un des deux textes pour l'entrainement, et l'autre pour l'évaluation (Figure 4.27). Il sera alors intéressant de comparer les performances obtenues pour les deux cas de la Figure 4.27, considérant les différences de contenu (style, lexique) entre ces deux romans.

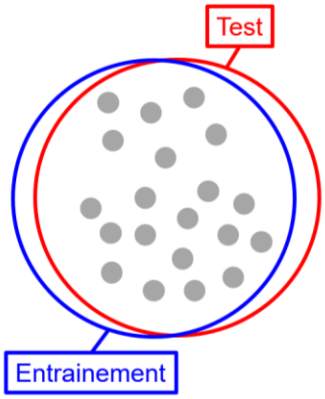


Figure 4.25 (a) : Cas non recommandé où on utilise le même jeu de données pour l'entrainement (cercle bleu) et le test (cercle rouge)

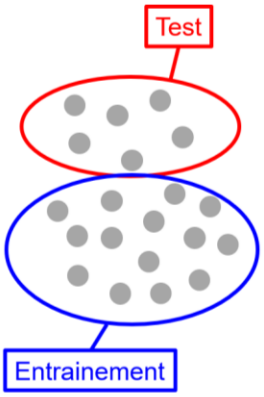


Figure 4.25 (b) : Cas recommandé où on utilise deux jeux de données indépendants pour l'entrainement et le test (ici, dans un rapport 70:30)

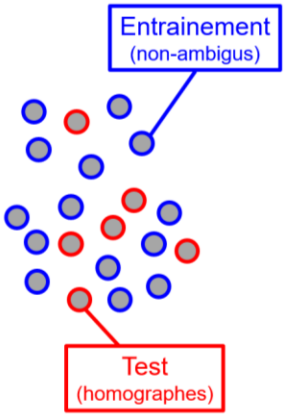


Figure 4.25 (c) : Cas particulier de l'entrainement automatique. Les mots non-ambigus (contours bleus) servent à l'entrainement, tandis que les homographes (contours rouges) servent au test

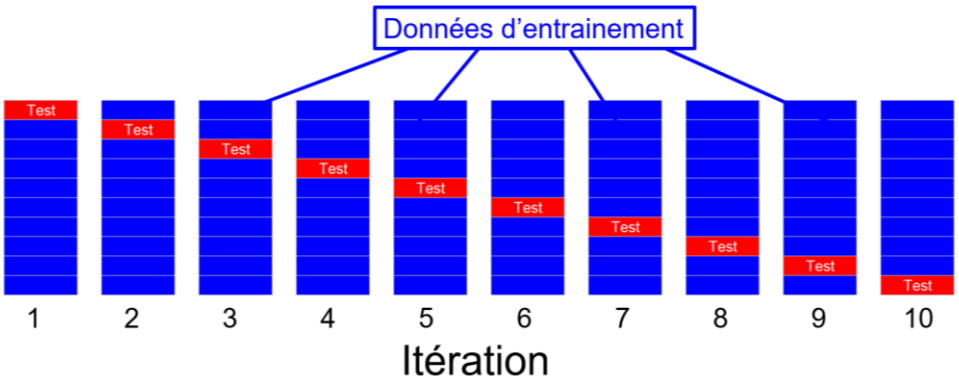


Figure 4.26 : Illustration du modèle « *k-fold* » pour l'évaluation d'algorithmes de classification. On sépare le jeu de données complet en *k* sous-ensembles (ici *k*=10). On effectue *k* itérations, où chaque sous-ensemble sert à tour de rôle de jeu de test et les *k*-1 ensembles restants servent à l'entrainement. La performance du modèle se calcule alors selon la moyenne des *k* performances

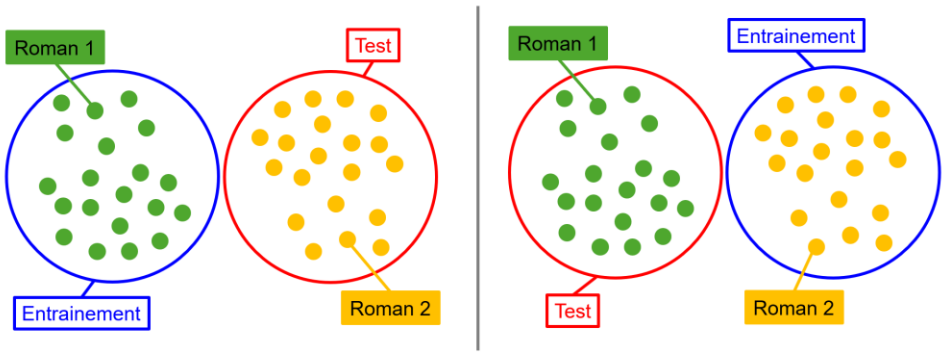


Figure 4.27 : Utilisation de deux corpus de référence distincts pour procéder à l'évaluation de l'algorithme de désambiguïsation, pour maintenir deux jeux de données (entrainement et évaluation) indépendants

4.6.4.2. Matrices de confusion

Une fois un algorithme de classification mis en place, il faut en mesurer la performance pour s'assurer qu'il puisse par la suite être appliqué avec confiance. La matrice de confusion est l'outil adopté pour ce projet pour comparer les prédictions de l'algorithme de désambiguïsation des homographes avec les classes réelles. Comme cet outil est couramment utilisé, il ne sera pas expliqué ici très en profondeur. On se contentera d'en expliquer l'usage spécifique à ce projet.

Dans un cas où un homographe peut appartenir à deux classes grammaticales différentes, on utilise une matrice de confusion à deux lignes et deux colonnes. Les lignes de la matrice correspondent aux classes réelles, tandis que les colonnes correspondent aux classes prédites par l'algorithme. Par exemple, l'homographe « place » peut être soit un verbe ou soit un nom commun. Au moment de la classification du mot « place », il y a quatre scénarios possibles, correspondant aux quatre cases de la matrice de confusion illustrée à la Figure 4.28. Si dans le contexte de la phrase le mot « place » est un verbe, et que l'algorithme prédit correctement que le mot est effectivement un verbe, on incrémente la valeur contenue à la première ligne et à la première colonne de la matrice. Mais si l'algorithme prédit de façon erronée que le mot devrait être un nom commun alors qu'il est réellement un verbe, c'est plutôt à la deuxième colonne de la première ligne qu'on incrémente la valeur de la matrice. L'opération est la même si le mot « place » dans le contexte d'une autre phrase est un nom commun, sauf qu'on considère alors la deuxième ligne de la matrice.

La matrice de la Figure 4.28 s'attarde à la performance de désambiguïsation d'un mot en particulier (« place »). Mais on peut en généraliser l'usage en y incorporant les résultats pour tous les homographes de type « verbe-nom commun ». Une telle matrice quantifie la capacité de l'algorithme à distinguer les formes verbales des noms communs. D'autres matrices semblables peuvent ensuite être bâties pour comparer entre elles les prédictions pour toutes les paires de classes grammaticales. De telles matrices de confusion comparant toutes les paires de classes grammaticales seront présentées au Chapitre 5.

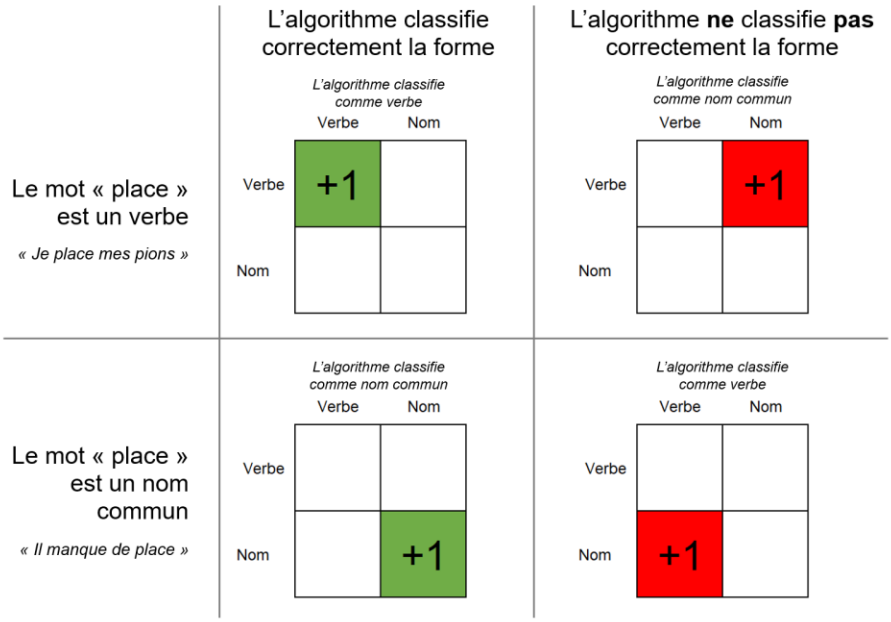


Figure 4.28 : Illustration du concept de matrice de confusion. Application à un homographe n'offrant que deux possibilités de classes grammaticales : le mot « place » peut être soit un verbe, soit un nom commun

Cependant, comme on l'a plus d'une fois mentionné, certains homographes offrent plus de deux possibilités de classes grammaticales. Par exemple, le mot « ferme » peut être ou bien un verbe, un adjectif ou un nom commun. Dans le cadre de ce projet, la performance de l'algorithme de désambiguïsation pour de tels homographes sera évaluée en suivant la méthode présentée à la Figure 4.29. On constate sur cette figure qu'à chaque fois que l'algorithme traite un homographe à trois classes, deux matrices de confusion doivent être mises à jour simultanément. Les matrices à mettre à jour dépendent de la classe réelle du mot dans le contexte de la phrase, ainsi que du succès ou non de la prédiction par l'algorithme. Une approche semblable est utilisée pour les mots offrant plus de trois possibilités (le mot « point » en offre quatre, le mot « tout » en offre cinq). Dans ces cas, davantage de matrices doivent être mises à jour à chaque fois que le mot « point » ou le mot « tout » est désambiguïsé.

Il est aussi possible de construire des matrices de confusion à plus de deux lignes et deux colonnes. Par exemple, au Chapitre 5, on présente des matrices de confusion globales comportant neuf lignes et neuf colonnes, correspondant aux neuf classes grammaticales considérées pour ce projet, offrant une vue d'ensemble de la performance de l'algorithme.

L'intérêt de la matrice de confusion, de façon générale, est qu'elle permet de quantifier la performance par l'intermédiaire de deux paramètres : la précision et le rappel. La précision est le pourcentage des cas où les cas réels sont bien prédits, tandis que le rappel est le pourcentage des cas où les prédictions sont bonnes. La distinction entre la précision et le rappel est pertinente, car de façon générale, une augmentation de la précision se fait au détriment du rappel, et vice-versa. On discutera davantage des matrices de confusion au Chapitre 5, au moment de présenter les résultats de la désambiguïsation.

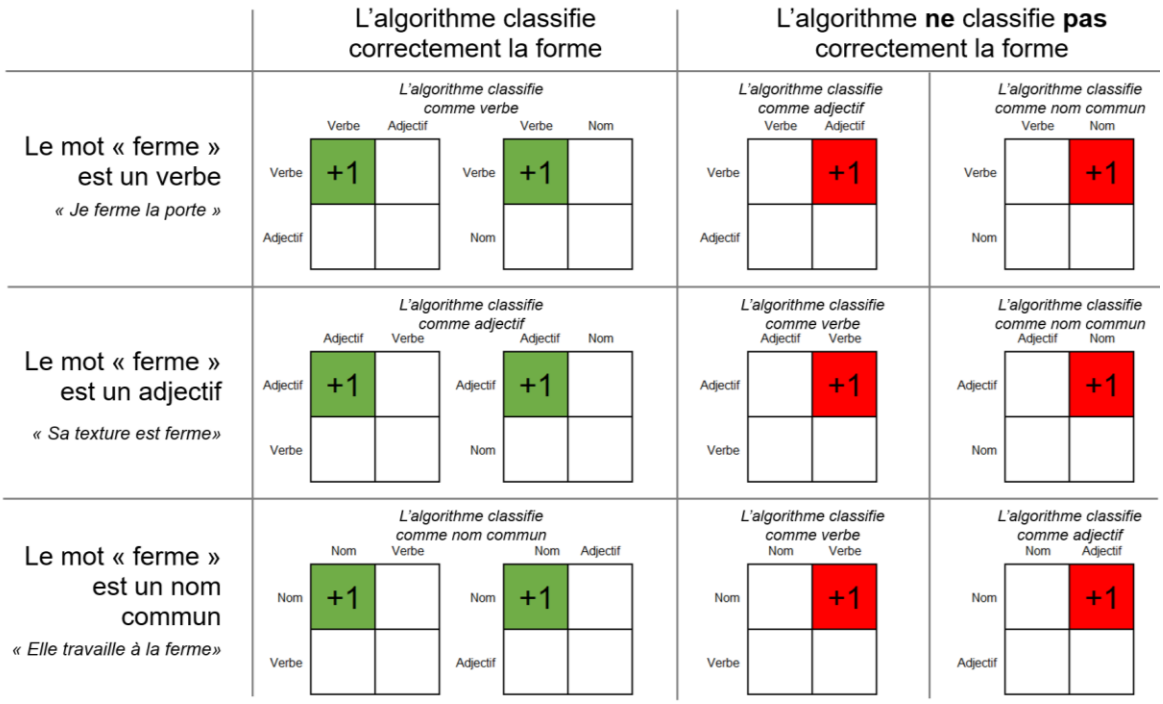


Figure 4.29 : Illustration du concept de matrice de confusion pour un cas d'homographe offrant trois possibilités de classes grammaticales : le mot « ferme » peut être soit un verbe, soit un adjectif, soit un nom commun

4.6.4.3. Analyse de la phrase « gauche-droite »

La désambiguïsation d'un homographe ne peut s'effectuer en considérant l'homographe seul, hors du contexte de son utilisation. En effet, seuls les liens qu'il entretient avec les autres mots de la phrase peuvent déterminer sa fonction grammaticale. L'opération de désambiguïsation des homographes doit donc s'effectuer en considérant la phrase dans son ensemble. On présente ici une première approche pour la désambiguïsation de tous les homographes d'une phrase, qu'on baptise l'approche « gauche-droite ». En suivant cette approche, on considère les homographes de la phrase un à la fois, dans l'ordre où ils apparaissent, donc de gauche à droite.

Tous les mots de la phrase qui précèdent le premier homographe sont non ambigus, par défaut. L'évaluation des caractéristiques de l'homographe dépendant de la classe grammaticale des mots le précédant se fait donc facilement. En revanche, les caractéristiques de l'homographe qui dépendent des mots suivants doivent tenir compte du fait que certains d'entre eux sont eux aussi des homographes. Dans un tel cas, l'approche « gauche-droite » considère chaque homographe situé à la droite de l'homographe sous étude comme appartenant simultanément à toutes les classes grammaticales possibles pour cet homographe. La même approche avait été utilisée pour l'entrainement automatique, tel que décrit à la Section 4.6.3.1. À mesure que chaque homographe de la phrase est désambiguïsé, l'ambiguïté « globale » de la phrase diminue, jusqu'au moment où le dernier homographe de la phrase (celui situé le plus à gauche) est finalement lui aussi désambiguïsé.

Pour illustrer le concept décrit plus haut où on assigne aux homographes situés à droite toutes leurs classes possibles en simultané, nous allons considérer deux caractéristiques en particulier, soit celles correspondant à la classe grammaticale du mot précédent, et à la classe grammaticale du mot suivant. Considérons par exemple la phrase suivante :

- « Cette demande ferme du procureur a été respectée. »

Au moment d'analyser l'homographe « demande », on doit déterminer quelle classe grammaticale correspond au mot précédent. Ici, le mot « cette » n'est pas un homographe, et l'information de la table de hachage « *tableRef* » nous informe que ce mot est un déterminant. On assigne donc la valeur de « 1 » à la caractéristique se rapportant au fait que le mot précédent est un déterminant. Le Tableau 4.17 illustre le détail des caractéristiques concernant le mot précédant le mot « demande » pour cette phrase. On y constate qu'on assigne 10 caractéristiques distinctes se rapportant au mot précédent, correspondant aux 9 classes grammaticales, ainsi qu'au cas où le mot est le premier de la phrase. Ces caractéristiques prennent la valeur de « 1 » (oui) ou « 0 » (non).

Tableau 4.17 : Caractéristiques concernant le mot *précédent*, évaluées pour le mot « demande » dans la phrase « Cette demande ferme du procureur a été respectée ». Seule la caractéristique « déterminant » prend la valeur « 1 », car le mot « cette » qui précède le mot « demande » est un déterminant

1er mot de la phrase	Verbe	Adjectif	Nom	Adverbe	Déterminant	Pronom	Préposition	Conjonction	Interjection
0	0	0	0	0	1	0	0	0	0

On doit ensuite déterminer la classe du mot suivant, soit le mot « ferme ». Ce mot peut être ou bien un verbe, un adjectif ou un nom commun. Comme il y a trois possibilités, on assigne alors la valeur de 1 à chacune des caractéristiques du mot suivant se rapportant aux classes verbe, adjectif et nom. Le Tableau 4.18 illustre le détail des caractéristiques concernant le mot suivant le mot « demande » pour cette même phra se. Tout comme pour le Tableau 4.17, 10 caractéristiques en tout sont associées au mot suivant, soit neuf pour les neuf classes grammaticales possibles, en plus du cas où le mot sous étude est le dernier de la phrase.

Tableau 4.18 : Caractéristiques pour le mot *suivant*, évaluées pour le mot « demande » dans la phrase « Cette demande ferme du procureur a été respectée ». Comme le mot suivant est « ferme » et que ce mot peut être ou bien un verbe, un adjectif ou un nom, ces trois caractéristiques prennent la valeur de 1

Verbe	Adjectif	Nom	Adverbe	Déterminant	Pronom	Préposition	Conjonction	Interjection	Dernier mot de la phrase
1	1	1	0	0	0	0	0	0	0

La même approche est utilisée pour toutes les autres caractéristiques. Prenons par exemple la caractéristique correspondant à la présence d'un participe passé après le mot sous étude. Considérons les deux phrases suivantes et attardons-nous à l'homographe « aura » :

- La journaliste **aura** fait son devoir.
- La journaliste **aura** un fait divers de plus à couvrir.

Si on évalue les caractéristiques correspondant à l'homographe « aura » dans ces deux phrases, et qu'on tente plus particulièrement d'évaluer la caractéristique « suivi d'un participe passé », cette caractéristique prendra la valeur de « 1 » dans les deux cas, puisque le mot « fait » *peut* être un participe passé. En effet, selon l'approche « gauche-droite », l'homographe « fait » n'est pas encore désambiguïsé au moment de l'analyse de l'homographe « aura ». Ainsi, l'algorithme ne peut pas encore déterminer si le mot « fait » est un nom commun, un participe passé ou encore un verbe conjugué. Comme il est donc *a priori* possible que le mot « fait » soit un participe passé, la caractéristique prend donc la valeur de « 1 », même si on sait que dans la deuxième phrase, le mot « fait » est un nom commun. On se retrouve donc, pour la deuxième phrase, avec une

caractéristique « erronée ». C'est là une faiblesse de l'approche gauche-droite dont on pourra mesurer les conséquences au Chapitre 5 au moment de l'évaluation de l'algorithme de désambiguïsation.

La Figure 4.30 résume le fonctionnement de l'approche « gauche-droite » pour la désambiguïsation des homographes, avec l'exemple d'une phrase tirée du roman « Le Rouge et le Noir ». On y constate qu'effectivement, on procède de gauche à droite pour la désambiguïsation, puisqu'à chaque itération, une nouvelle case verte avec une classe grammaticale unique s'ajoute du côté gauche. À chaque itération, toutes les classes possibles pour chaque homographe non encore désambiguïsé sont considérées. Celles-ci apparaissent au-dessus de chaque case.

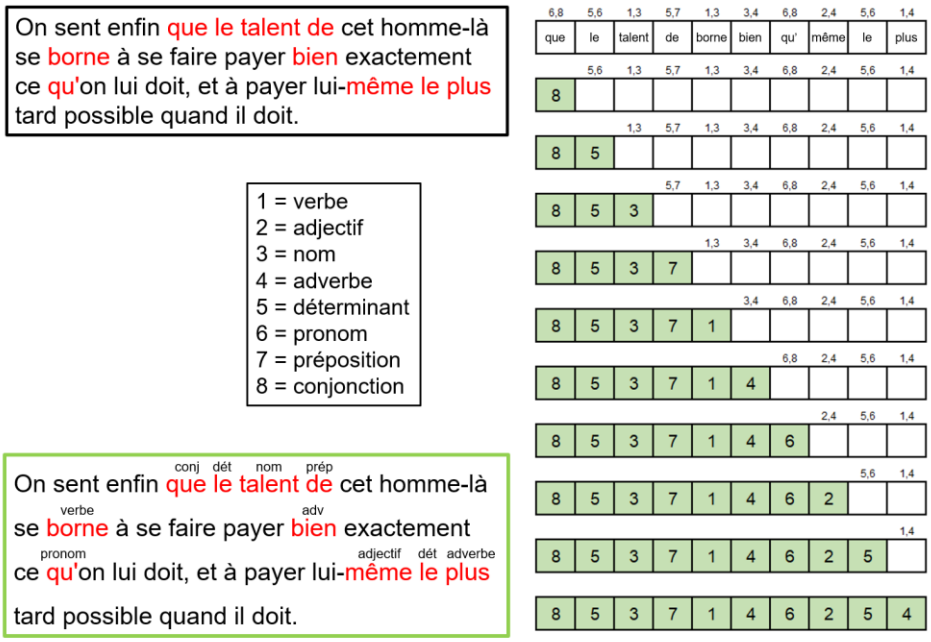


Figure 4.30 : Illustration de la désambiguïsation des homographes d'une phrase selon l'approche « gauche-droite », avec l'exemple d'une phrase tirée du roman « Le Rouge et le Noir ». Les homographes y sont identifiés en rouge. Les cases du tableau de droite reçoivent les codes des classes grammaticales, à mesure que les homographes sont désambiguïsés

4.6.4.4. Analyse globale de la phrase

Alors que pour l'approche « gauche-droite » décrite à la section précédente, on désambiguïse un homographe à la fois, procédant de gauche à droite dans la phrase, la deuxième approche considère plutôt tous les homographes simultanément, donc globalement. Pour y arriver, on répertorie d'abord toutes les combinaisons possibles de classes grammaticales correspondant aux homographes présents *avant* de procéder à toute désambiguïsation. Prenons par exemple la phrase suivante extraite du roman « Le Rouge et le Noir » :

- « Elle doit à cette admirable position une des **vues les plus** pittoresques **de** France. »

Cette phrase comporte quatre homographes (« vues », « les », « plus » et « de »), inscrits en rouge. Trois de ces homographes (« les », « plus » et « de ») comportent deux classes grammaticales possibles, tandis que « vues » offre trois possibilités. En tout, il y a donc 3 x 2 x 2 x 2 = 24 combinaisons possibles, toutes illustrées à la Figure 4.31.

Une fois toutes les possibilités répertoriées, on calcule un « score » pour chacune d'elles. On calcule ce score en multipliant la probabilité associée à chaque classe grammaticale pour chacun des mots. Par exemple, on calcule le score de la Possibilité 1 en multipliant quatre probabilités, puisque la phrase comporte quatre homographes. On calcule d'abord à la première colonne et à la première ligne de la Figure 4.31 la probabilité que le mot « vues » soit un verbe, en supposant que le mot « les » est un déterminant, le mot « plus » est un verbe et finalement le mot « de » un déterminant. Toujours à la première ligne, mais cette fois à la deuxième colonne, on calcule ensuite la probabilité que le mot « les » soit un déterminant, en supposant que le mot « vues » est un verbe, le mot « plus » est un verbe et finalement le mot « de » un déterminant. Et ainsi de suite, pour les autres homographes de la première ligne. On procède ensuite de la même façon pour chacune des 23 autres lignes.

On multiplie ensuite les quatre probabilités de chaque ligne entre elles pour obtenir le score correspondant à chacune des 24 possibilités de la Figure 4.31. La possibilité parmi les 24 obtenant le plus haut score est alors considérée comme la structure de phrase (combinaison de classes grammaticales) la plus probable. Dans l'exemple fourni, on note que la valeur maximale de score est obtenue pour la 21^{e} possibilité, correspondant aux codes de classes grammaticales « 3 », « 5 », « 4 » et « 7 ». L'algorithme considère donc que l'homographe « vues » est un nom commun, l'homographe « les » est un déterminant, l'homographe « plus » est un adverbe, et finalement l'homographe « de » est une préposition. Il est à noter qu'une approche alternative aurait consisté à calculer la *somme* des probabilités, plutôt que leur *produit*. Cette alternative est aussi illustrée à la Figure 4.31 (dernière colonne). On voit que dans ce cas précis, la prédiction se retrouve à être la même selon les deux approches (multiplication ou somme). Toutefois, pour ce projet, les scores ont été calculés sur la base de la multiplication, car on a ainsi obtenu de meilleurs résultats, comme on le démontrera au Chapitre 5.

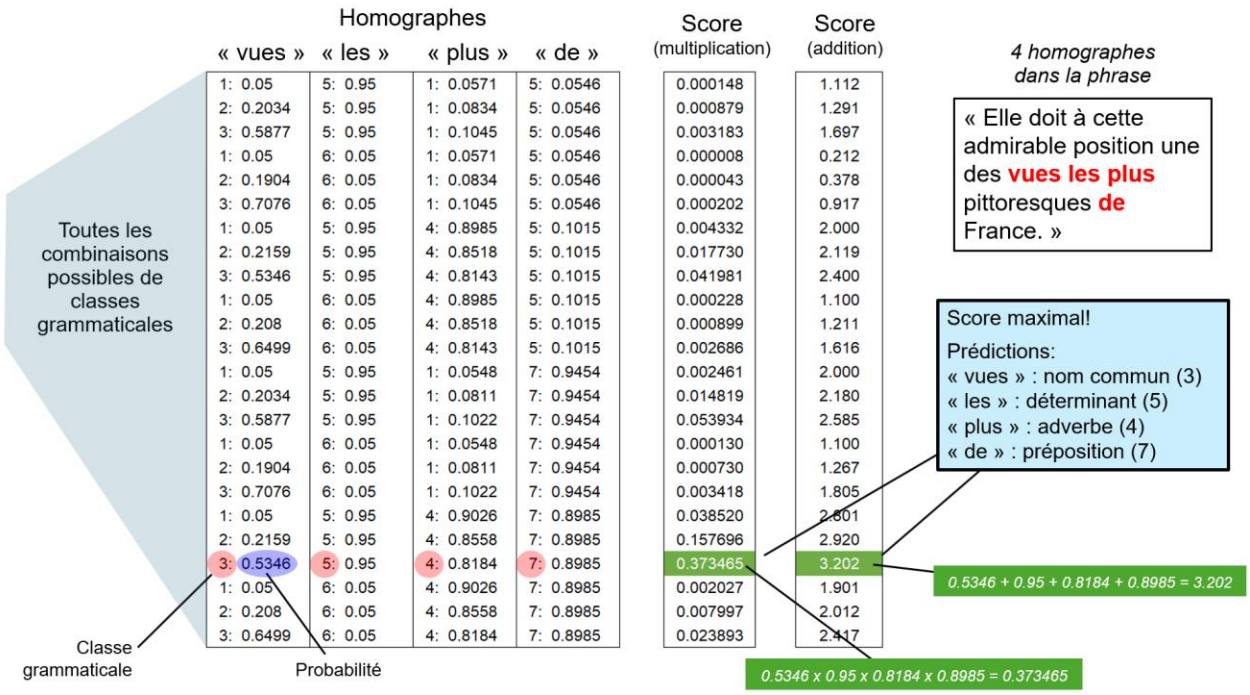


Figure 4.31 : Illustration du calcul du « score » pour chaque phrase désambiguïsée au complet pour l'algorithme d'analyse globale de la phrase

L'avantage de l'approche considérant l'analyse globale de la phrase est qu'on évite les erreurs dans les évaluations de caractéristiques qu'on avait observées pour l'approche « gauche-droite » résultant du fait que les homographes à la droite du mot sous étude n'étaient pas encore désambiguïsés. Il était en effet nécessaire de considérer en parallèle toutes les possibilités de classes grammaticales. En considérant plutôt tous les homographes en simultané comme on le fait ici avec l'approche globale, on met de côté toute ambiguïté, ce qui mène à une prédiction potentiellement plus précise.

En revanche, cette approche introduit le risque de devoir effectuer un très grand nombre d'itérations. Par exemple, une phrase contenant 10 homographes comportant chacun deux possibilités de classes grammaticales, engendre 2^{10} =1024 itérations. C'est donc dire qu'il faut effectuer 1024 itérations pour cette phrase, contrairement à l'approche gauche-droite qui s'effectuerait en 20 itérations (10 x 2). De surcroît, certaines phrases comportent bien plus de 10 homographes. En effet, la phrase du roman « Le Rouge et le Noir » comportant le plus d'homographes en compte 34. Cette phrase est reproduite ici, avec les homographes en rouge :

> « Leur croissance rapide et leur belle verdure tirant sur le bleu, ils la doivent à la terre rapportée, que monsieur le maire a fait placer derrière son immense mur de soutènement, car, malgré l'opposition du conseil municipal, il a élargi la promenade de plus de six pieds (quoiqu'il soit ultra et moi libéral, je l'en loue), c'est pourquoi dans son opinion et dans celle de monsieur Valenod, l'heureux directeur du dépôt de mendicité de Verrières, cette terrasse peut soutenir la comparaison avec celle de Saint-Germain-en-Laye. »

Ces 34 homographes offrent tous deux classes grammaticales possibles, sauf l'homographe « fait » qui en compte trois, puisqu'il peut être un verbe, un adjectif (participe passé employé seul) ou un nom commun. Ainsi, le nombre total d'itérations à considérer pour cette phrase est calculé de la façon suivante :

$$2^{33} \times 3 = 2.57689 \times 10^{10}$$

Ce très grand nombre de possibilités se retrouve à ralentir l'exécution de l'algorithme, puisque toutes les possibilités doivent être explorées afin d'identifier celle qui est optimale. En fait, ce nombre est si élevé qu'il faudrait sans doute un ordinateur hyperpuissant pour arriver à considérer toutes les possibilités. La Figure 4.32 illustre l'effet exponentiel du nombre d'homographes dans une phrase sur le nombre total de combinaisons d'homographes possibles. Cette approche globale de la désambiguïsation des homographes d'une phrase, sous cette forme du moins, ne peut donc pas être adoptée. Il faut explorer une alternative.

En s'attardant de nouveau à la phrase ci-haut tirée du roman « Le Rouge et le Noir », on peut facilement remarquer que la classe du premier homographe a bien peu de chances d'influencer la classe du tout dernier homographe de la phrase. En effet, cette phrase est en fait composée de plusieurs propositions subordonnées, n'ayant en bout de ligne peu de liens grammaticaux entre elles. On émettra donc l'hypothèse qu'un homographe n'est influencé grammaticalement que par les mots situés dans son environnement rapproché. On choisira donc, arbitrairement, de ne faire varier en simultané pas plus de cinq homographes « voisins » à la fois. Cette hypothèse, comme on va le voir, va nous permettre de réduire considérablement le nombre d'itérations à effectuer.

L'algorithme proposé, qu'on peut définir comme l'approche globale modifiée, va comme suit :

- On applique en premier lieu l'approche « gauche-droite » de la Section 4.6.4.3 pour assigner une classe grammaticale initiale à chaque homographe. On émet l'hypothèse que l'algorithme « gauche-droite » nous offre ainsi un point de départ convenable, obtenu rapidement
- On se concentre ensuite sur les 5 premiers homographes de la phrase. Pour ces 5 homographes, on ignore les prédictions de l'approche « gauche-droite » et on répertorie plutôt toutes les combinaisons possibles de classes grammaticales pour ces 5 mots. En supposant 2 classes grammaticales par homographe, on compte 32 possibilités (2^5).
- On calcule ensuite un « score » pour ces 32 possibilités, en tenant compte aussi des classes grammaticales assignées par la méthode « gauche-droite » pour les autres homographes de la phrase (au-delà des 5 premiers). Chaque score se calcule en multipliant les probabilités associées à chaque homographe, selon le calcul de régression
- On repère ensuite parmi les 32 possibilités celle qui a donné le score le plus élevé.
- On assigne ensuite aux 5 homographes les 5 classes grammaticales correspondant à ce score le plus élevé.
- On « avance » ensuite d'un homographe. C'est-à-dire qu'au lieu de considérer les 5 premiers, on considère maintenant les homographes du deuxième jusqu'au sixième. Le premier homographe conserve la classe grammaticale qu'on vient de lui assigner. Et les homographes au-delà du sixième conservent encore pour l'instant les valeurs assignées par l'approche « gauche-droite »
- On répertorie toutes les combinaisons possibles de classes grammaticales pour les homographes du deuxième jusqu'au sixième et on calcule le score associé à chacune des possibilités, comme on l'a fait précédemment.
- Une fois de plus, on repère la combinaison donnant le score maximal, et on assigne les 5 homographes aux classes correspondant à ce score maximal.
- On avance encore d'un homographe, et on répète le processus, jusqu'à ce que le dernier homographe de la phrase corresponde au cinquième homographe du sous-ensemble

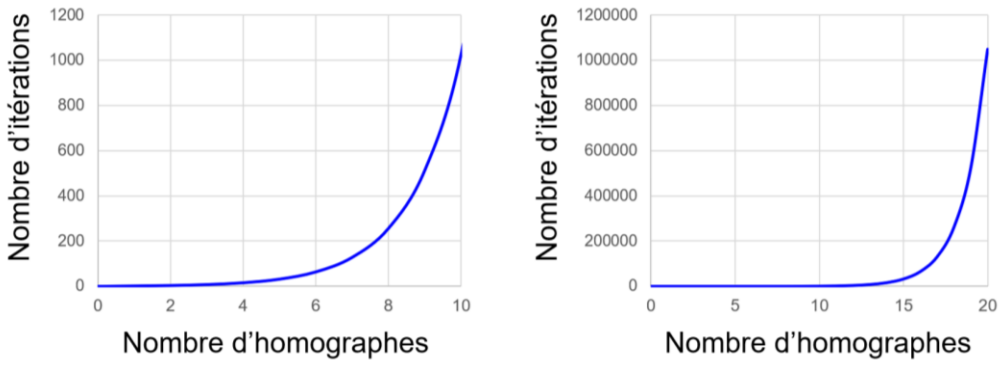


Figure 4.32 : Graphique du nombre de combinaisons possibles de classes grammaticales en fonction du nombre d'homographes dans une phrase, en supposant 2 classes possibles par homographe. À gauche, l'axe horizontal se rend à 10 homographes, correspondant à 1024 itérations. À droite, on constate qu'avec 20 homographes, le nombre d'itérations atteint plus d'un million

On peut donc associer cette nouvelle approche à un « glissement » progressif sur l'ensemble des homographes de la phrase. Cette approche est illustrée à la Figure 4.33. Le nombre total d'itérations nécessaires pour désambiguïser tous les homographes de la phrase dépend à la fois du nombre d'homographes dans la phrase, et du nombre de classes grammaticales possibles pour chaque homographe. Pour simplifier les calculs, supposons pour l'instant que chaque homographe ne comporte que deux possibilités de classes grammaticales.

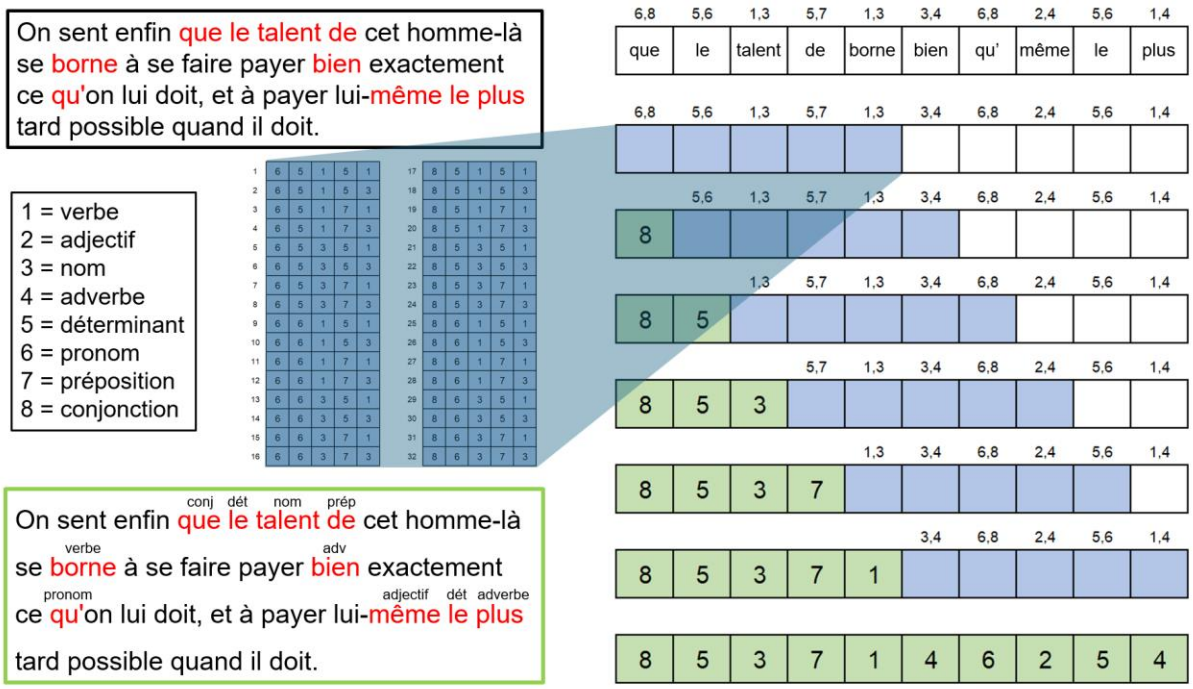


Figure 4.33 : Illustration de la désambiguïsation des homographes d'une phrase selon l'approche « globale » avec glissement, avec l'exemple d'une phrase tirée du roman « Le Rouge et le Noir ». Les homographes y sont identifiés en rouge. Les cases en bleu du tableau de droite correspondent aux 5 homographes considérés simultanément à chaque itération (32 possibilités en général). Les cases vertes reçoivent les codes des classes grammaticales, à mesure que les homographes sont désambiguïsés

Il y a deux cas à considérer, selon le nombre d'homographes dans la phrase :

- **5 homographes ou moins dans la phrase**. Dans ce cas, l'approche globale modifiée est équivalente à l'approche globale initiale. Le nombre d'itérations se calcule selon :

  $$\#itérations = 2^n$$

  où n est le nombre d'homographes dans la phrase. Ainsi, avec 5 homographes, on doit considérer 32 itérations.

- **Plus de 5 homographes dans la phrase**. Dans ce cas, il faut calculer le nombre d'itérations en considérant le nombre de « glissements » de 5 homographes

  $$\#itérations = 2^5 \times (n-4) = 32 \times (n-4)$$

  où n est le nombre d'homographes dans la phrase. Ainsi, avec par exemple 34 homographes, on doit considérer 960 itérations. Cet algorithme est de complexité algorithmique $\theta(n)$ (linéaire)

En fait, dans le cas précis de la phrase du roman « Le Rouge et le Noir » citée plus haut, le nombre total d'itérations serait de 1040, en considérant le fait qu'un des homographes comporte trois possibilités de classes grammaticales, au lieu de 2. Ces 1040 itérations pour cette approche modifiée sont bien plus raisonnables que les milliards d'itérations requises pour cette même phrase en utilisant l'approche globale initiale. Et comme cet exemple correspond à la phrase comportant le plus d'homographes, on peut en conclure qu'on aura rarement besoin de plus de 1000 itérations pour complètement désambiguïser chacune des phrases du roman.

La Figure 4.34 compare graphiquement le nombre d'itérations requises en fonction du nombre d'homographes dans la phrase selon les trois approches décrites plus haut. Pour ce mémoire, on ne considèrera donc que deux de ces trois approches, soit l'approche « gauche-droite » (la plus rapide) et l'approche « globale modifiée avec glissement ».

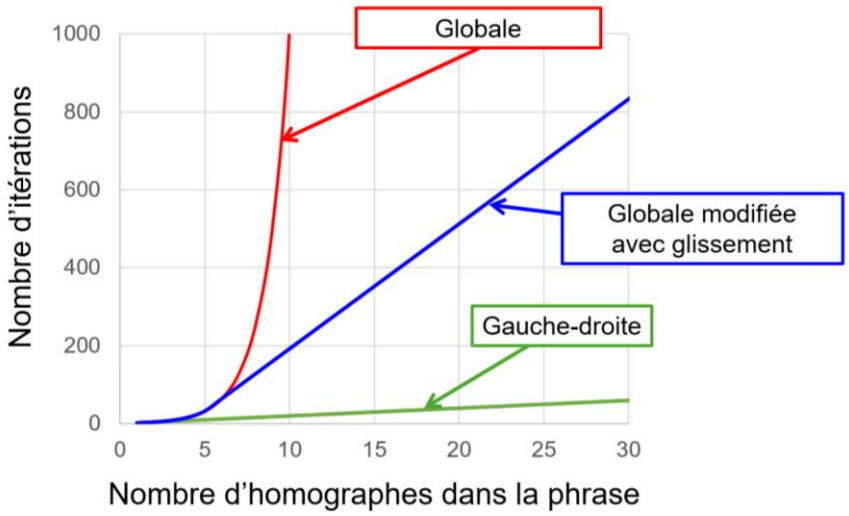


Figure 4.34 : Graphique du nombre d'itérations requises en fonction du nombre d'homographes dans une phrase, en supposant deux classes grammaticales possibles par homographe. Comparaison des trois approches discutées

4.6.5. Tests spécialisés de désambiguïsation

L'algorithme de classification par apprentissage machine discuté jusqu'à présent cherche à déterminer à laquelle des neuf classes grammaticales appartient chaque homographe. Comme les mêmes caractéristiques (« *features* ») sont évaluées pour tous les mots, l'hypothèse sous-jacente est que tout mot appartenant à une classe grammaticale donnée est interchangeable avec tout autre mot issu de la même classe. Il en résulte qu'à chaque classe grammaticale correspond une « signature » distincte en fonction des caractéristiques, permettant de distinguer entre elles ces neuf classes.

Hélas, comme on le verra au Chapitre 5 portant sur les résultats de la lemmatisation, certains homographes ne sont pas désambiguïsés avec autant de succès que la plupart des autres homographes issus de la même classe. Les raisons précises pour ce plus faible taux de réussite ne sautent pas toujours aux yeux, mais on peut supposer que ces homographes ne sont possiblement pas utilisés dans exactement les mêmes circonstances que la moyenne des autres. Ou encore que la combinaison de classes de ces homographes est différente. Par exemple, il est peut-être plus facile de désambiguïser un verbe pouvant aussi être un nom (exemple, « demande »), qu'un verbe pouvant aussi être un adverbe (exemple : « plus »).

Face à de tels obstacles, une méthode additionnelle a été mise au point pour ce projet, aussi basée sur l'apprentissage machine. En effet, des tests « spécialisés » ont été mis en place pour certains de ces homographes généralement mal désambiguïsés. Chaque test spécialisé se concentre sur un mot en particulier. Mais un tel test spécialisé n'est pertinent que si l'homographe en question est fréquent dans le texte. Cette condition tient au fait que l'apprentissage machine n'est efficace qu'en présence de suffisamment de données brutes. De plus, il va de soi que plus un homographe mal désambiguïsé apparait souvent dans le texte, plus il vaut la peine de s'y attaquer.

Le fonctionnement de ces tests spécialisés est simple. En premier lieu, on explore un ensemble de caractéristiques spécifiques jugées pertinentes pour désambiguïser chacun de ces homographes posant problème. On ne fait pas ici nécessairement appel aux mêmes caractéristiques générales décrites jusqu'à présent. De nouvelles caractéristiques peuvent être ajoutées. Dans certains cas, un petit nombre de caractéristiques s'est avéré suffisant. Au moment de l'entrainement, l'algorithme génère alors une banque de données distincte pour chacun de ces homographes « spécialisés ». Par exemple, pour l'homographe « tout », pour lequel un test spécialisé a été mis en place, la banque de données se retrouve à contenir autant d'observations que le nombre d'occurrences du mot « tout » dans la partie du texte lemmatisée manuellement. Un algorithme de régression linéaire logistique est ensuite effectué, de la même façon qu'on l'a décrit dans le cas général (Section 4.6.2.2), afin de déterminer les paramètres de l'équation de régression. Au moment de l'évaluation, à chaque fois que le lemmatiseur rencontre le mot « tout », il fait appel au test spécialisé pour désambiguïser ce mot, plutôt qu'au test « général ». Au Chapitre 5, on mentionnera exactement les quelques homographes pour lesquels on a créé de tels tests spécialisés, on discutera des caractéristiques utilisées, et on indiquera l'amélioration de la performance de désambiguïsation que ces tests ont apportée.

Il va de soi que les tests spécialisés ne peuvent être « entrainés » que par la méthode manuelle, puisque par défaut, tous les mots utilisés pour l'entrainement automatique ne sont *pas* des homographes.

4.6.6. Algorithme de « *guesser* »

Tel que mentionné à la Section 3.1.3, des banques de données ont été bâties manuellement pour ce projet, permettant de générer tous les mots inclus dans les deux corpus de référence, qu'on a inclus à la table de hachage « *tableRef* ». Il a été mentionné que compte tenu de la portée limitée de ce projet, il n'a pas été envisagé d'inclure à ces banques de mots *tous* les mots de la langue française, ce qui aurait représenté une tâche colossale.

Mais comme on n'a pas inclus tous les mots de la langue dans les fichiers texte, on peut évidemment s'attendre, lors de la lemmatisation d'un nouveau texte, à ce que l'algorithme se butte à l'occasion à un mot non encore inclus dans ses banques de données. Dans un tel cas, l'algorithme ne dispose donc pas d'indice de départ pour ce mot « inconnu », en ce qui concerne les classes grammaticales possibles pour ce mot. Mais tout n'est pas perdu, car un algorithme a été mis en place pour ce projet, afin de deviner, sur la base des caractéristiques générales développées pour le projet, la classe grammaticale la plus probable pour chaque mot non classé. Cet algorithme, appelé « *guesser* », fonctionne de la même façon que l'algorithme général de désambiguïsation discuté à la Section 4.6.4. La seule différence est qu'on considère ici la possibilité que chaque mot inconnu puisse appartenir à n'importe laquelle des neuf classes grammaticales. Il en découle qu'on ne peut s'attendre à une performance de désambiguïsation aussi élevée pour ces mots non classés que dans le cas de la majorité des homographes auxquels on n'associe que deux classes grammaticales possibles. La Figure 4.35 illustre le fonctionnement de l'algorithme de « *guesser* ». L'algorithme, sur la base des équations de régressions logistiques, calcule toutes les probabilités par paires de classes. Par exemple, on calcule la probabilité que le mot soit un adjectif par opposition à un nom commun (probabilité de 0.061 au tableau de la Figure 4.35). On multiplie ensuite toutes ces probabilités pour chacune des classes grammaticales (chaque ligne de la Figure 4.35) pour arriver à un score pour chaque classe. La classe recevant le score le plus élevé est ensuite celle qu'on assigne au mot non classé. Cette méthode est identique à celle présentée plus haut à la Section 4.6.2.2.2, sauf pour la taille du tableau qui est plus grande dans le cas du « *guesser* » (9 par 9). Il est à noter que chaque élément de la diagonale prend la valeur de « 1 ».

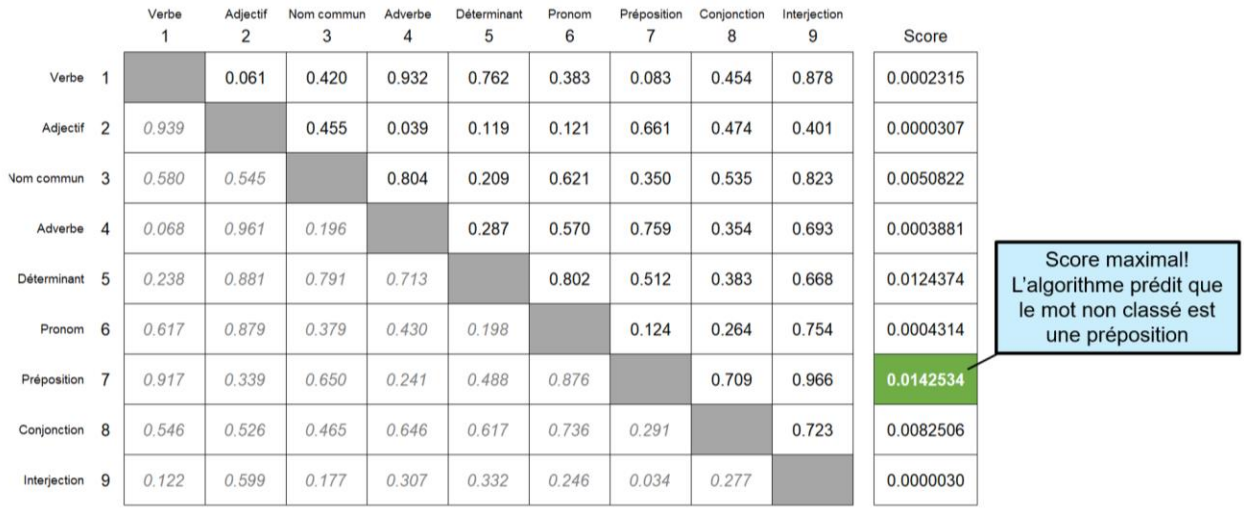


Figure 4.35 : Illustration du fonctionnement de l'algorithme « *guesser* » pour assigner la classe grammaticale la plus probable à un mot non inclus dans la table de hachage « *tableRef* »

4.7. Au-delà de l'apprentissage machine

L'apprentissage machine se fonde sur des analyses statistiques. Dans le cas qui nous concerne, si une certaine caractéristique est rencontrée pour un homographe, cette caractéristique nous donne un *indice* quant à la classe grammaticale possible de l'homographe. Il n'est pas ici question de certitude. Mais puisque plusieurs caractéristiques sont évaluées simultanément pour chaque homographe, on peut arriver à une probabilité raisonnable pour chaque mot en combinant tous ces indices. En apprentissage machine, tout est donc question de « probabilité ».

Mais dans certaines situations, il s'est avéré plus judicieux d'appliquer des règles déterministes pouvant soit renverser la décision de l'apprentissage machine, ou soit permettre de mieux l'orienter, en éliminant d'emblée certaines possibilités. Trois règles déterministes, ne faisant donc pas appel à l'apprentissage machine, ont été incluses à l'algorithme de désambiguïsation et sont décrites plus bas.

4.7.1. Analyse des participes passés

Tel qu'on l'a mentionné à la Section 4.1.4, on a considéré pour ce projet tous les participes passés comme étant à la fois des formes verbales et des adjectifs. En effet, les participes passés « employés seuls » (sans auxiliaire « être » ou « avoir ») se comportent exactement comme des adjectifs. Ainsi, par défaut, tous les participes passés se retrouvent à être des homographes pour ce projet, et doivent en conséquence être désambiguïsés.

Mais en plus d'être considérées à la fois comme verbes et adjectifs, certaines formes de participes passés se retrouvent aussi comme autres formes verbales conjuguées. Par exemple :

- Les examens que j'ai *réussis*. (« réussis » ici est un participe passé)
- Tu *réussis* ton examen. (« réussis » ici est une forme conjuguée qui n'est pas un participe passé)

Certaines autres formes de participes passés peuvent aussi être associées à d'autres classes grammaticales. Par exemple :

- Il a *été* choisi. (« été » est ici un participe passé)
- L'*été* est ma saison préférée (« été » ici est un nom commun)

Finalement, certains participes passés peuvent être à la fois une forme verbale qui n'est pas un participe passé, et un mot appartenant à une autre classe grammaticale. Par exemple :

- J'ai *fait* mon devoir. (« fait » est un participe passé)
- Il *fait* son devoir. (« fait » ici est une forme conjuguée qui n'est pas un participe passé)
- C'est un *fait* accepté de tous. (« fait » ici est un nom commun)

On peut donc classifier les participes passés en quatre catégories ou « scénarios » distincts, qu'on résume au Tableau 4.19. En observant ce tableau, on constate que le scénario le plus simple pour la désambiguïsation est le Scénario 1, puisqu'il n'implique que deux possibilités. Les Scénarios 2 et 3 sont caractérisés par trois possibilités. Finalement, le Scénario 4 est le plus complexe, puisqu'il implique quatre possibilités.

Tableau 4.19 : Les 4 scénarios possibles de participes passés

	Scénario 1	Scénario 2	Scénario 3	Scénario 4
Participe passé avec auxiliaire	✓	✓	✓	✓
Participe passé employé seul (adjectif)	✓	✓	✓	✓
Verbe conjugué non participe passé		✓		✓
Autre classe grammaticale			✓	✓
Exemples	« mangé »	« réussis »	« été »	« fait »

L'algorithme détermine donc en premier lieu à quel « scénario » d'homographe on a affaire. La façon d'y arriver est simple, car il suffit de compiler les classes grammaticales présentes parmi les éléments de la liste chaînée de la table de hachage « *tableRef* ». Si on n'y retrouve que les classes 1 (verbe) et 2 (adjectif), et que la seule forme verbale possible est un participe passé (temps de verbe = 10) il est question du Scénario 1. Si on a en plus affaire à un verbe conjugué à un temps autre que participe passé (temps de verbe non égal à 10), il est alors question du Scénario 2. Si aucune forme verbale autre que « participe passé » n'est possible, mais qu'une classe grammaticale autre que 1 (verbe) ou 2 (adjectif) est présente, il est alors question du Scénario 3. Finalement, si un des éléments est un verbe conjugué autre que participe passé et qu'en plus on retrouve l'homographe à une classe grammaticale autre que 1 et 2, il est alors question du Scénario 4.

Une fois le scénario identifié, l'approche employée pour faciliter la désambiguïsation des homographes pouvant être des participes passés vise à déterminer si l'homographe en question a plus de chances d'être un adjectif que tout autre classe grammaticale (« test adjectif »). Ce test ne considère donc pas si l'homographe a plus de chance d'appartenir à une classe grammaticale plutôt qu'à une autre n'impliquant *pas* le cas « adjectif ». Les indices employés sont les suivants :

- Premier mot de la phrase (indice adjectif)
- Directement précédé d'un point d'interrogation (indice pas un adjectif)
- Directement suivi d'un trait d'union (indice pas un adjectif)
- Auxiliaire « avoir » situé devant (indice pas un adjectif)
- Directement précédé d'un pronom personnel (indice pas un adjectif)
- Verbe attributif situé devant, sauf le verbe « être » (indice adjectif)
- Forme pronominale, donc présence de « me », « te », « t' », « se », ou « s' » devant, mais aussi présence du verbe « être » devant (indice pas un adjectif)
- Directement précédé d'un nom commun, mais mauvais accord (indice pas un adjectif)
- Directement précédé d'un nom commun bien accordé (entrecoupé ou non d'un adverbe d'intensité) (indice adjectif)
- Participe passé courant se conjugue avec l'auxiliaire « être » (indice pas un adjectif)
- Précédé du verbe « avoir », puis du participe passé « été » (indice adjectif)

La Figure 4.36 illustre le fonctionnement de l'algorithme pour les participes passés. Dans le cas du Scénario 1 (S1), le résultat du « test adjectif » détermine de façon déterministe la classe grammaticale de l'homographe (classe « 1 » ou classe « 2 »).

Dans le cas du Scénario 2 (S2), si le « test adjectif » est négatif, on peut donc assigner de façon certaine que l'homographe est un verbe, car les deux possibilités restantes (participe passé ou verbe conjugué) correspondent toutes deux à la classe « 1 ». En revanche, si le « test adjectif » est positif, on doit tenir compte du fait que l'homographe pourrait aussi être un verbe conjugué. Comme il reste encore ces deux options, on doit se rabattre sur l'apprentissage machine pour prendre la décision finale.

Pour les Scénarios 3 et 4, (S3 et S4), le « test adjectif », si négatif, permet d'éliminer la possibilité d'adjectif (classe « 2 ») de la liste chaînée, ce qui simplifie quelque peu la situation. Une méthode Java a été bâtie pour modifier la liste chaînée pour en retirer toutes les options contenant la classe « 2 » (extrait de code 4.6, où on fournit en entrée la classe « 2 »). Mais en bout de ligne, pour les scénarios S3 et S4, que le test soit positif ou négatif, on n'a pas le choix de se rabattre sur l'apprentissage machine pour prendre la décision finale, car il reste inévitablement au moins deux classes possibles.

L'algorithme déterministe pour les participes passés ne sert donc finalement, qu'à potentiellement exclure la possibilité que l'homographe puisse être un participe passé employé seul (adjectif). Une fois cette détermination faite, il est possible que la classe grammaticale soit automatiquement déterminée (Scénario 1 ou un des deux cas du Scénario 2). Dans tous les autres cas, il faut ensuite se rabattre à au modèle général d'apprentissage machine pour la désambiguïsation, avec possiblement une liste chaînée réduite, s'il a été déterminé que l'homographe ne peut être un adjectif.

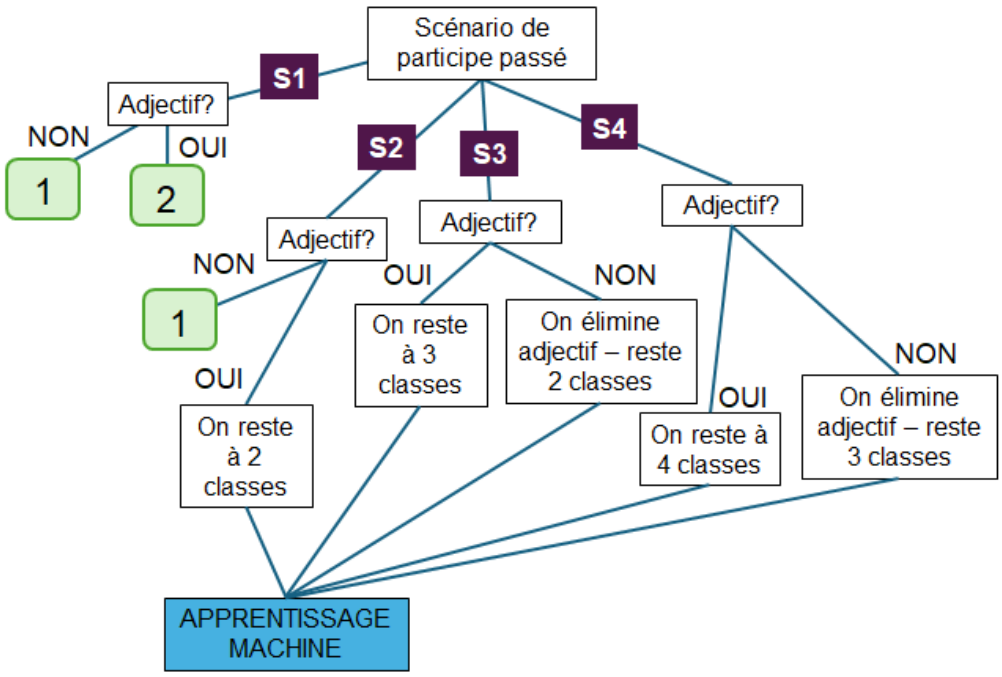


Figure 4.36 : Illustration de l'algorithme déterministe servant à simplifier les homographes impliquant des participes passés. Les cases vertes correspondent aux classes prédites de façon déterministes (1 = verbe, 2 = adjectif)

Extrait de code 4.6 : Méthode pour retirer une classe grammaticale donnée (*POSred*) d'un objet-mot (*motObjetOriginal*).

```
public static ObjetTableRef EnleverPOSunique(ObjetTableRef motObjetOriginal,
                                              int POSred){

int POSCourant, POSequiv;
int tailleListe = motObjetOriginal.liste.size();
LinkedList<LemmeObjet> motObjet = new LinkedList<>();

for (int k=0;k<tailleListe;k++){
    LemmeObjet lemmeObjet;
    lemmeObjet=motObjetOriginal.liste.get(k);
    POSCourant=lemmeObjet.GetPOS();
    POSequiv=Ut.Simple(POSCourant);
    if(POSequiv!=POSred){ // On garde le lemme, on l'ajoute à motObjet
        motObjet.add(lemmeObjet);
    }
} // fin du for k pour vérifier si on garde le lemme

motObjetOriginal.liste=motObjet;
return motObjetOriginal;
}
```

4.7.2. Analyse des locutions

A la Section 3.2.5, on a mentionné que l'algorithme de désambiguïsation développé pour ce projet recherche la présence de certaines locutions ou syntagmes impliquant la combinaison d'une forme verbale suivie d'un nom commun. Cette vérification est pertinente dans le cas où le nom commun en question est en fait un homographe. La présence devant ce nom d'une forme verbale d'un infinitif en particulier peut en effet parfois nous informer sur la classe grammaticale de l'homographe.

A la Section 3.2.5, certains de ces noms communs avaient été identifiés, tels que « place », « compte », « part », « note », « grâce » et plusieurs autres. L'algorithme de désambiguïsation, lorsqu'il rencontre un de ces homographes, vérifie si le mot en question est précédé d'une forme verbale spécifique issue d'un syntagme. À chaque nom commun de la liste, on associe le ou les verbes spécifiques menant à un syntagme pour ce nom. Par exemple, pour le mot « place », on vérifie si ce mot n'est pas précédé du verbe « prendre » (comme dans « j'ai pris place ») ou du verbe « faire » (comme dans « nous ferions place »). Mais comme la forme verbale peut être conjuguée à n'importe quel temps et n'importe quelle personne, il y a plusieurs formes verbales à vérifier. Cette vérification est grandement simplifiée (Figure 4.37) en s'attardant au « lemme » du verbe, donc son « infinitif », plutôt qu'à sa forme fléchie (conjuguée). Le lemme du verbe est facilement accessible, puisque emmagasiné par l'algorithme de lemmatisation.

Le fonctionnement de l'algorithme est donc simple : pour chaque nom commun faisant potentiellement partie d'un tel syntagme, on vérifie s'il est précédé d'une forme verbale dont l'infinitif (le lemme) est associé à ce nom commun. Si tel est le cas, on en conclut que l'homographe suivant le verbe est un nom commun, indépendamment de ce que l'algorithme d'apprentissage machine aurait pu prédire. L'extrait de code 4.7 fournit la portion de l'algorithme d'identification des syntagmes se rapportant à l'homographe « place ».

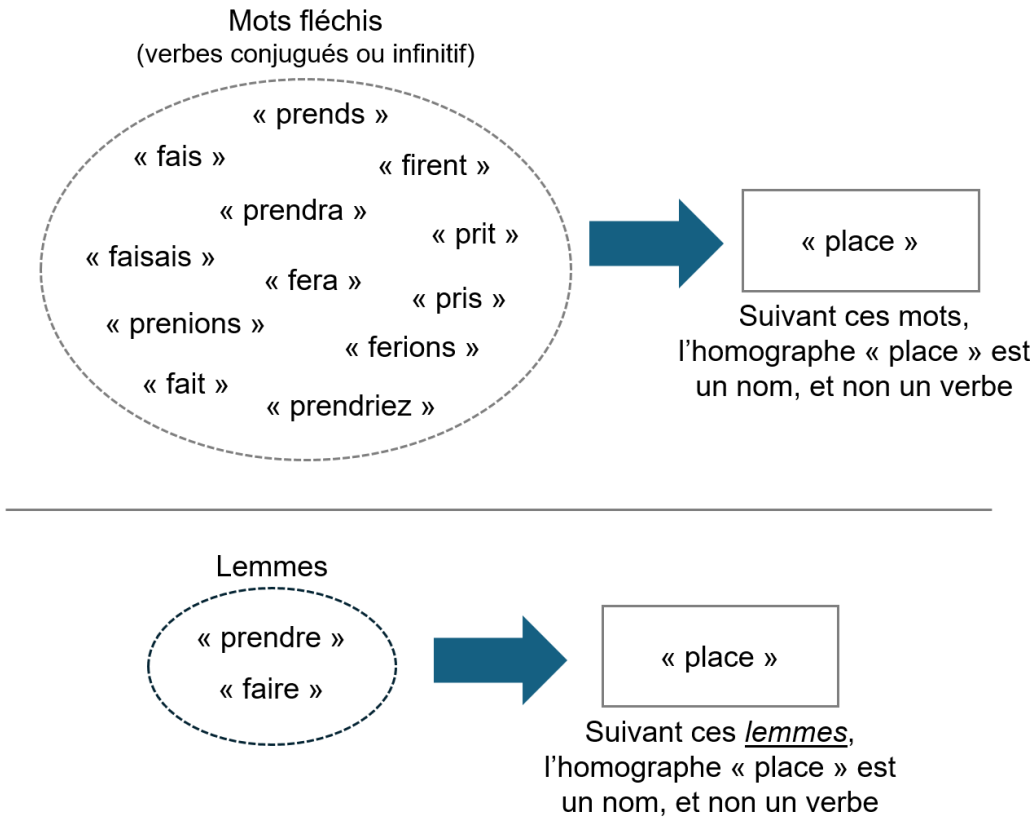


Figure 4.37 : Illustration du bénéfice de vérifier l'infinitif (le lemme) d'un verbe pour identifier un syntagme, plutôt que de devoir vérifier toutes les formes conjuguées du verbe

Extrait de code 4.7 : Extrait de la méthode d'identification des syntagmes se rapportant précisément à l'homographe « place ». Le vecteur « *infVerbes* » contient le lemme (infinitif) des autres mots de la phrase. Lorsque le mot « place » est précédé de l'infinitif « prendre » ou de l'infinitif « faire », le vecteur « *POSOut* » qui détermine la classe grammaticale du mot « place » est assigné à une valeur de « 3 » (nom commun)

```
if(mots[i].equals("place")){ // prendre place, faire place: "place" est un nom
    if(infVerbes[i-1].equals("prendre")|infVerbes[i-1].equals("faire")){
        POSOut[i]=3;
        POSVecteur[i][1]=0;
        verbe[i]=false;
    }
}
```

4.7.3. Analyse des statistiques sur les verbes

Jusqu'à présent, les analyses présentées au Chapitre 4 se sont toutes limitées à la phrase dans laquelle on retrouve l'homographe, sans déborder de ce contexte. Mais tel que mentionné à la Section 3.2.6, l'algorithme mis en place considère aussi des statistiques sur les verbes utilisés dans l'*ensemble du corpus* pour faciliter la désambiguïsation d'homographes pouvant être des formes verbales. On a ainsi présenté à la Section 3.2.6 deux tests visant à évaluer la probabilité qu'un tel homographe puisse en effet être un verbe. Le fait qu'on s'attarde ainsi aux verbes tient du fait que les formes verbales renferment de façon générale davantage d'information que les mots d'autres classes grammaticales. En effet, tandis qu'il n'existe que quatre formes possibles d'un nom ou d'un adjectif (combinaisons possibles du genre et du nombre), il existe pour la plupart des verbes jusqu'à 51 formes possibles, en combinant les temps du verbe (une dizaine) et la personne du verbe (6 possibilités). Les deux tests décrits au Chapitre 3 tirent donc profit de cette information pour aider à repérer les formes verbales.

En sortie, ces deux tests fournissent un paramètre semblable, soit un facteur quantifiant la possibilité que l'homographe soit un verbe. S'il est considéré, à la lumière de chacun de ces tests, que le mot a de bonnes chances d'être un verbe, le facteur est assigné à la valeur maximale « 1 ». Si au contraire, on considère que le mot n'a que peu de chances d'être un verbe, on assigne à ce facteur une valeur (positive) inférieure à « 1 ». La valeur en question dépend du niveau de « confiance » par rapport à la prédiction. On calcule ensuite un facteur statistique global, qu'on obtient en multipliant entre eux les facteurs obtenus pour ces deux tests.

Ce facteur est par la suite utilisé lors de l'application de l'algorithme d'apprentissage machine. En effet, on se rappelle que la régression logistique binaire donne en sortie une valeur comprise entre zéro et un. Dans le cas où un homographe peut être un verbe, on multiplie la valeur obtenue par la régression par le facteur statistique décrit plus haut, baptisé *testFreq* pour ce projet. On se retrouve ainsi à réduire la probabilité que l'homographe soit déterminé comme étant un verbe, lorsqu'on multiplie le résultat de la régression par un facteur plus petit que « 1 ». Ces tests statistiques ne sont donc pas déterministes : ils ne font qu'influer sur la probabilité qu'un homographe soit classifié comme un verbe. Autrement dit, même s'il était déterminé, selon ces deux tests statistiques, que l'homographe n'est probablement pas un verbe, l'algorithme d'apprentissage machine pourrait néanmoins en conclure le contraire. Une telle situation peut se produire si les caractéristiques générales de l'apprentissage machine s'apparentent à la « signature » d'une forme verbale.

Les algorithmes et paramètres utilisés pour ces deux tests sont décrits plus en détail aux sections suivantes. Dans les deux cas, il faut au préalable disposer des statistiques sur toutes les formes verbales incluses dans le texte. On présente donc en premier lieu, à la Section 4.7.3.1, comment celles-ci sont compilées.

4.7.3.1. Compilation des statistiques des verbes

On crée en premier lieu une table de hachage *statVerbe* dont la clé est l'infinitif d'un verbe (son lemme) et dont la valeur est un vecteur d'entiers comprenant 51 éléments. Ces éléments correspondent aux 51 formes verbales discutées plus haut, et représentées au Tableau 4.20. Par l'intermédiaire de cette table de hachage, on fait donc le décompte de toutes les formes verbales possibles de tous les homographes pouvant être des verbes. Ce processus de compilation des formes verbales est illustré à la Figure 4.38. On y donne l'exemple du mot « demande », un homographe pouvant représenter ou bien un nom commun, ou un verbe. En tout, cinq formes verbales sont associées au mot « demande », soit la première et la troisième personnes du présent de l'indicatif, la première et la troisième personnes du subjonctif présent, ainsi que la deuxième personne de l'impératif présent. On voit comment ces cinq formes verbales

s'incorporent à la fois à la table de hachage ainsi qu'au vecteur global. Cet algorithme de compilation de statistiques de verbe s'effectue en parallèle avec la lemmatisation de base du texte (Section 4.3). À la fin de la lemmatisation de base, on se retrouve donc avec la table de hachage *statVerbe* complète comprenant l'information sur tous les homographes du texte pouvant être des formes verbales, ainsi qu'avec le vecteur global. Cette table de hachage ainsi que le vecteur global servent ensuite à l'évaluation des deux tests présentés aux Sections 4.7.3.2 et 4.7.3.3.

Tableau 4.20 : Les codes (indices) associés aux 51 formes verbales

Code	Forme	Code	Forme	Code	Forme
0	Prés 1ere pers sing	17	Passé S 3e pers plur	34	Subj Imp 2e pers plur
1	Prés 2e pers sing	18	Futur S 1ere pers sing	35	Subj Imp 3e pers plur
2	Prés 3e pers sing	19	Futur S 2e pers sing	36	Impératif 2e pers sing
3	Prés 1ere pers plur	20	Futur S 3e pers sing	37	Impératif 1ere pers plur
4	Prés 2e pers plur	21	Futur S 1ere pers plur	38	Impératif 2e pers plur
5	Prés 3e pers plur	22	Futur S 2e pers plur	39	Cond 1ere pers sing
6	Imp 1ere pers sing	23	Futur S 3e pers plur	40	Cond 2e pers sing
7	Imp 2e pers sing	24	Subj P 1ere pers sing	41	Cond 3e pers sing
8	Imp 3e pers sing	25	Subj P 2e pers sing	42	Cond 1ere pers plur
9	Imp 1ere pers plur	26	Subj P 3e pers sing	43	Cond 2e pers plur
10	Imp 2e pers plur	27	Subj P 1ere pers plur	44	Cond 3e pers plur
11	Imp 3e pers plur	28	Subj P 2e pers plur	45	Participe présent
12	Passé S 1ere pers sing	29	Subj P 3e pers plur	46	Part passé masc sing
13	Passé S 2e pers sing	30	Subj Imp 1ere pers sing	47	Part passé masc plur
14	Passé S 3e pers sing	31	Subj Imp 2e pers sing	48	Part passé fém sing
15	Passé S 1ere pers plur	32	Subj Imp 3e pers sing	49	Part passé fém plur
16	Passé S 2e pers plur	33	Subj Imp 1ere pers plur	50	Infinitif

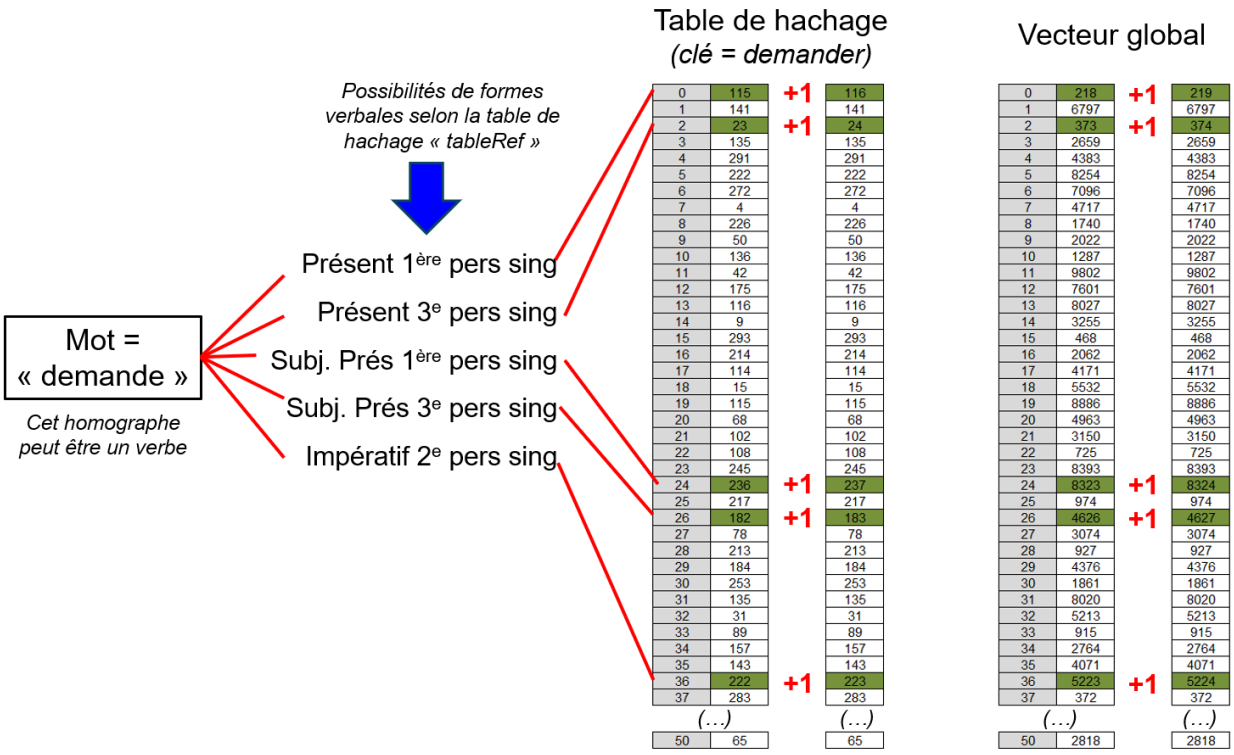


Figure 4.38 : Illustration du fonctionnement de l'algorithme de compilation des statistiques de verbe avec valeurs fictives. Les éléments correspondant aux indices des 5 formes verbales correspondant au mot « demande » sont incrémentés de 1

4.7.3.2. Premier test statistique – formes verbales limitées

Comme on le mentionnait plus haut, il existe plusieurs formes verbales (51), mais peu de formes de verbes et d'adjectifs (au plus, quatre). Et pour les noms communs « non-animés » (tels que décrits au Tableau 4.5) il n'existe en fait que deux formes possibles, soit le singulier ou le pluriel. Ce premier test statistique tire justement profit de cet écart de nombre possible de formes *fléchies* entre les verbes et les autres classes grammaticales. En présence d'un homographe pouvant être un verbe, on extrait toutes les formes verbales distinctes observées dans le corpus correspondant à l'infinitif de ce verbe (selon la table de hachage « *tableRef* »). Si on n'en dénombre qu'une seule, il est fort à parier que l'homographe en question n'est pas un verbe, puisque dans un long texte, on retrouve plus souvent qu'autrement un verbe sous plus d'une forme. On se demande ensuite si l'homographe peut être un nom commun ou un adjectif, selon la table de hachage « *tableRef* ». Si c'est le cas, et si on ne retrouve l'homographe que sous deux formes, et que ses deux formes peuvent être toutes les deux un nom ou un adjectif, et qu'en plus une forme est le pluriel de l'autre, il est plus probable que l'homographe ne soit *pas* un verbe. Ce processus est illustré à la Figure 4.39. La programmation d'un tel algorithme est simple, car elle n'implique qu'une série d'énoncés de type « IF » faisant appel à l'information incluse à la table de hachage *statVerbe* décrite à la section précédente. Il suffit ensuite d'assigner une valeur pour le paramètre *testFreq* en fonction des différents cas illustrés à la Figure 4.39. Pour les cas où on estime que l'homographe est probablement un verbe, on assigne la valeur maximale de « 1 ». Il reste ensuite deux valeurs à déterminer pour ce paramètre : quand une seule forme du verbe est présente, et quand seulement deux formes sont présentes et que l'une est le pluriel de l'autre (en supposant qu'il s'agisse d'un nom commun ou d'un adjectif). Les valeurs accordées au paramètre dans ces deux cas ont été optimisées pour maximiser l'efficacité de l'algorithme de désambiguïsation. Il en sera question au Chapitre 5, quand on évaluera l'efficacité et la pertinence de ce test statistique.

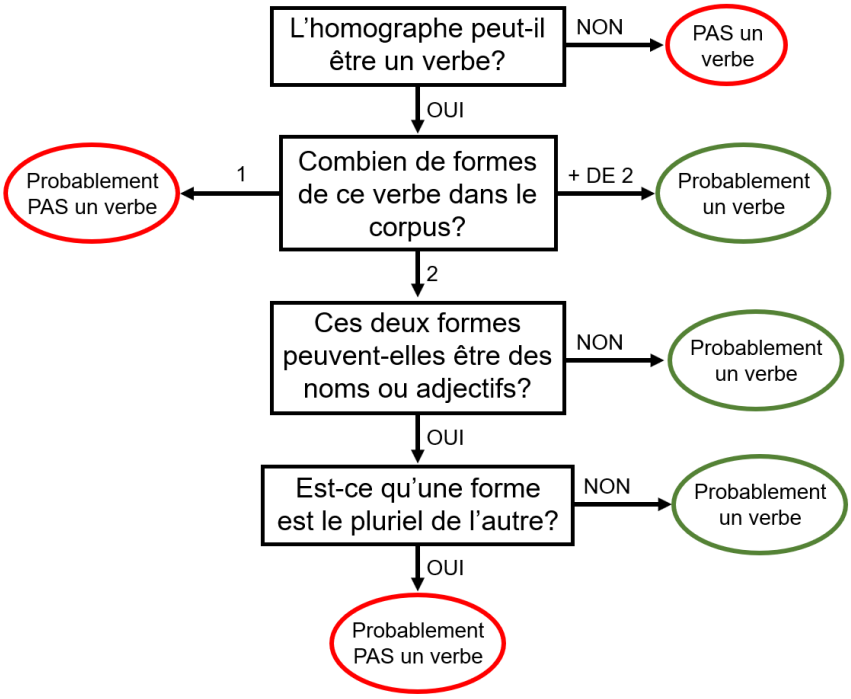


Figure 4.39 : Fonctionnement du premier test statistique aidant à désambiguïser les homographes comportant une forme verbale

4.7.3.3. Deuxième test statistique – ratio des formes verbales

Comme on le décrivait à la Section 3.2.6, le deuxième test statistique pour les verbes consiste à calculer le ratio d'occurrences de la forme verbale homographique (combinaison de temps et de personne) sur le nombre total d'occurrences de ce verbe, puis de comparer ce même ratio calculé cette fois sur la base de tous les autres verbes du corpus. Un ratio passablement plus élevé pour le verbe sous analyse en comparaison avec ce même ratio pour l'ensemble des verbes, fournit un indice comme quoi l'homographe en question n'est probablement *pas* un verbe.

L'exemple fourni au Chapitre 3 concernait l'homographe « plus », qui peut être ou bien un adverbe ou bien un verbe. Quand l'algorithme de désambiguïsation rencontre l'homographe « plus », il calcule en premier lieu le ratio du nombre d'occurrences du mot « plus » sur le nombre total d'occurrences du verbe « plaire » (l'infinitif de « plus ») dans le corpus entier. Appelons ce ratio le « ratio 1 ». L'algorithme identifie par la suite les formes verbales (temps et personnes) associées à l'homographe « plus ». On parle ici de la première personne et de la deuxième personne du singulier, au passé simple (« je plus », « tu plus »), correspondant aux codes « 12 » et « 13 » du Tableau 4.18. L'algorithme extrait ainsi le nombre total d'occurrences de ces deux formes verbales dans le texte *pour tous les verbes* autres que « plaire » et calcule le ratio de ces formes verbales sur le nombre total de formes verbales dans le texte (excluant « plaire »). Appelons ce ratio le « ratio 2 ». Ce processus est illustré à la Figure 4.40, avec des valeurs fictives pour le verbe « plaire » et pour tous les autres verbes.

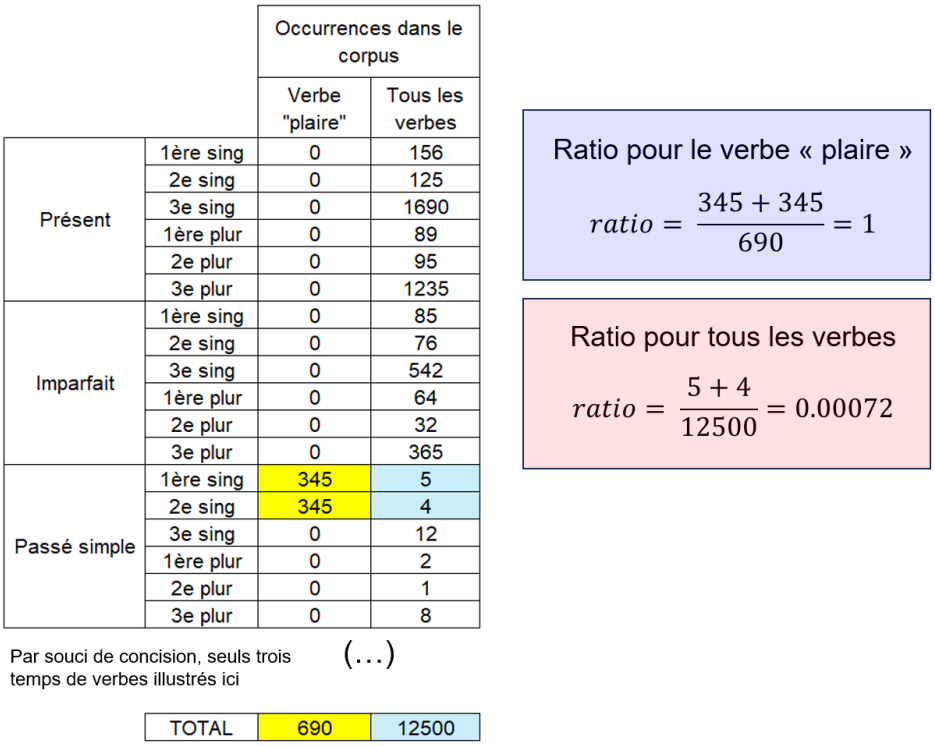


Figure 4.40 : Fonctionnement du deuxième test statistique aidant à désambiguïser les homographes comportant une forme verbale, ici appliqué à un exemple fictif relié à l'homographe « plus ». Dans ce cas, le ratio pour le verbe « plaire » est beaucoup plus élevé que pour l'ensemble des verbes du corpus. On en déduit que le mot « plus » n'est probablement pas un verbe dans ce cas précis

La comparaison entre le « ratio 1 » et le « ratio 2 » s'effectue ensuite en calculant un troisième ratio, le « ratio final », qui s'obtient en divisant le « ratio 1 » par le « ratio 2 » :

$$ratio\ final = \frac{ratio\ 1}{ratio\ 2}$$

On constate pour cet exemple fictif, que le « ratio 1 » pour le verbe « plaire » est très élevé en comparaison avec le « ratio 2 » portant sur tous les autres verbes du corpus. La valeur de « ratio final » est donc très élevée. Une valeur élevée de « ratio final » offre un fort indice comme quoi le mot « plus » dans cet exemple fictif a peu de chances d'être un verbe. Bien que les chiffres de la Figure 4.40 soient fictifs, le « ratio final » est typiquement élevé pour le mot « plus », puisque l'adverbe « plus » est couramment utilisé dans les textes français, alors que les formes au passé simple du verbe « plaire » le sont beaucoup moins.

Pour mettre cet algorithme statistique en place, deux paramètres doivent être déterminés. Tout d'abord, on doit déterminer une valeur-seuil pour le « ratio final » au-delà de laquelle on détermine que l'homographe a peu de chances d'être une forme verbale. Le deuxième paramètre à déterminer est le facteur *testFreq* par lequel on multipliera le résultat de la fonction de régression logistique, pour diminuer la probabilité que l'apprentissage machine conclue que l'homographe est un verbe. Ces deux paramètres ont été optimisés pour maximiser l'efficacité de l'algorithme de désambiguïsation. Il en sera question au Chapitre 5, quand on évaluera l'efficacité et la pertinence de ce test statistique.

4.8. Résumé du fonctionnement de l'algorithme de désambiguïsation

On a vu à la Section 4.6 que l'algorithme de désambiguïsation développé pour ce projet repose principalement sur l'apprentissage machine. En effet, plusieurs caractéristiques grammaticales ont été définies, et différentes approches ont été proposées autant pour l'entrainement que l'évaluation de modèles d'apprentissage machine. Mais on a aussi vu à la Section 4.7 qu'en plus de l'algorithme d'apprentissage machine, d'autres algorithmes se sont greffés à l'analyse, ceux-ci étant de nature « déterministe » à l'opposé de l'apprentissage machine axé sur les statistiques. Dans le cas des participes passés, on en fait l'analyse en amont de l'apprentissage machine, dans le but de réduire les possibilités à évaluer. Dans le cas des statistiques sur les verbes du corpus, on s'en sert pour modifier le résultat de l'apprentissage machine avant d'effectuer l'algorithme. Finalement, on repère la présence de syntagmes *après* avoir effectué l'apprentissage machine, si bien que ce test final peut parfois se retrouver à « renverser » la décision prise au préalable par l'algorithme d'apprentissage machine. Afin d'offrir une vue d'ensemble du processus complet de désambiguïsation, la Figure 4.41 présente toutes ces étapes. Les étapes d'apprentissage machine sont indiquées par des rectangles au fond bleu. Le résultat final de la désambiguïsation d'un mot est fourni dans un carré au contour vert. On se rend compte que plusieurs « chemins » différents peuvent mener à une décision, dépendamment de la nature du mot sous analyse. Le processus de la Figure 4.41 est effectué pour chacun des homographes du corpus, au moment de la désambiguïsation.

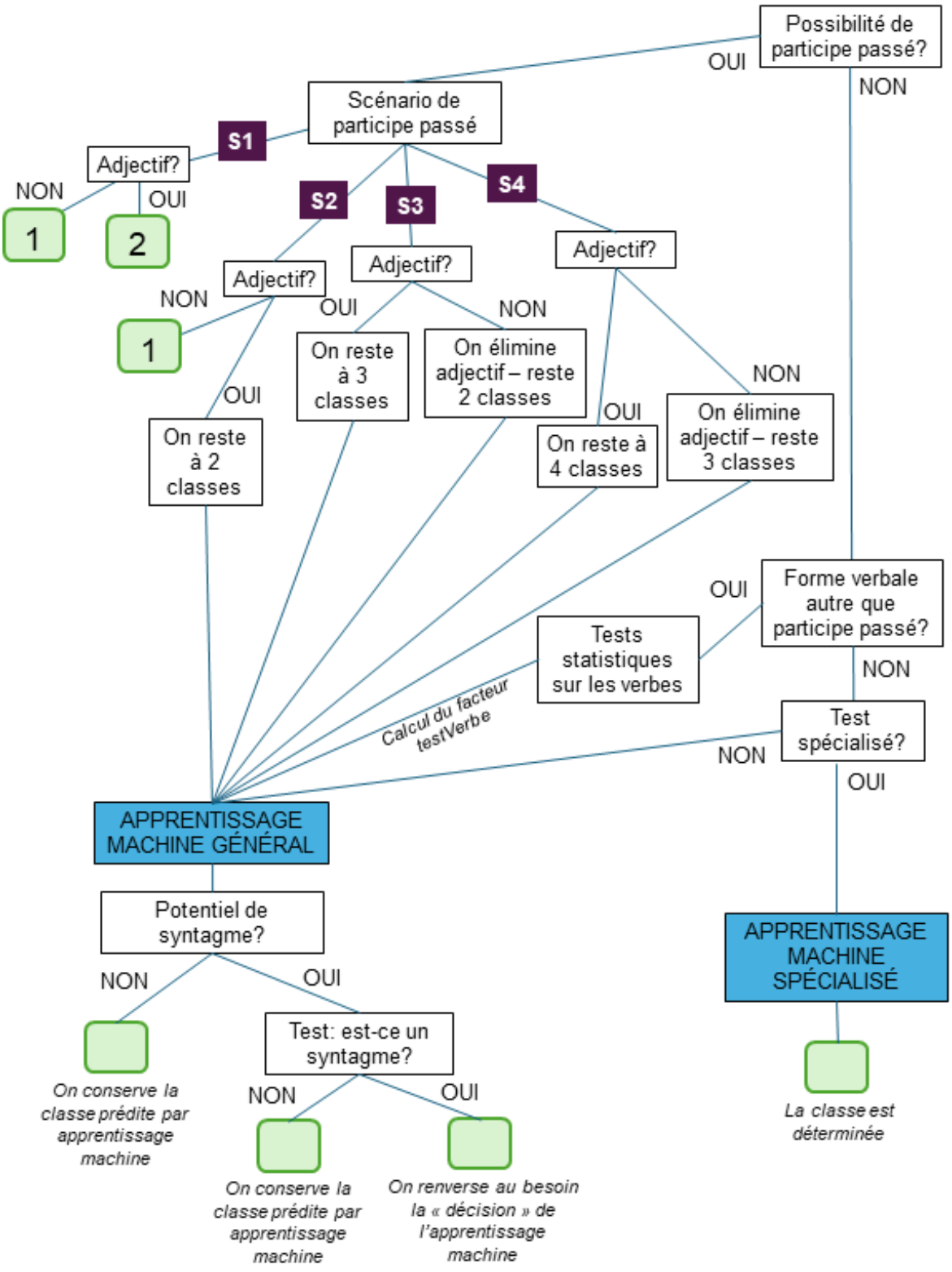


Figure 4.41 : Illustration du fonctionnement de l'algorithme de désambiguïsation des homographes pour ce projet

4.9. Information en sortie de l'outil de lemmatisation (Étape 1)

Une fois la lemmatisation du corpus de référence complétée, ce qui inclut les étapes de désambiguïsation des homographes, un fichier texte est automatiquement généré pour fournir le sommaire des résultats, pour analyse subséquente par un *humain,* au besoin. Les informations fournies en sortie comprennent :

- Nombre de phrases
- Nombre de mots
- Distribution du nombre de mots par phrase
- Nombre de lemmes distincts
- Fréquences des classes grammaticales
- Fréquences des temps et des personnes de verbe
- Présence d'homographes distincts selon leurs classes grammaticales
- Fréquences des homographes selon leurs classes grammaticales
- Détail des phrases dans lesquelles tous les homographes sont retrouvés, et dans le cas de l'entrainement manuel, efficacité de la désambiguïsation pour chaque mot individuel

Au Chapitre 5, on retrouve cette information détaillée en lien avec les corpus de référence utilisés pour ce projet.

Mais en plus de ces informations fournies dans un fichier texte à l'utilisateur, comme référence, l'algorithme de lemmatisation doit fournir en sortie une foule d'informations nécessaires à la génération de textes aléatoires automatiquement lemmatisés, à l'Étape 2. Ces informations sont fournies sous forme de valeurs de variables et d'objets, passés aux classes et méthodes de l'Étape 2. Les informations principales passant des algorithmes de l'Étape 1 à ceux de l'Étape 2 ont été décrites à la Section 3.2.1

4.10. Étape 2 : Algorithmes pour la génération de textes aléatoires

L'Étape 2, qui consiste à générer aléatoirement des textes automatiquement lemmatisés, est le cœur de ce projet. L'Étape 1 ne sert finalement qu'à fournir l'information de base pour effectuer l'Étape 2, qui a été résumée à la Section 4.9. Comme le texte lemmatisé se bâtit phrase par phrase, l'algorithme de base est celui qui bâtit les phrases. Le texte final est ensuite généré tout simplement en joignant bout à bout chacune des phrases indépendamment formées et qui n'ont donc pas de lien entre elles. Tout ce que ces phrases ont en commun est que les mots qui les composent ont été extraits du même lexique de base généré à l'Étape 1.

Chaque phrase est évidemment composée de mots se suivant dans un ordre précis. Mais comme on veut que cette phrase soit lemmatisée, c'est donc dire que pour chacun de ses mots, il faut aussi fournir un lemme, selon le même ordre. Et puis finalement, pour faciliter la comparaison avec des outils de lemmatisation existants comme on le verra au Chapitre 6, on cherche aussi à accoler une étiquette morpho-syntaxique à chaque mot de la phrase. Les informations incluses sur les étiquettes varient selon la classe grammaticale. Par exemple, pour un nom commun, on fournit le genre et le nombre, tandis que pour un verbe on doit mentionner son temps et sa personne.

Mais la phrase se bâtit par étapes, tel qu'on l'a mentionné à la Section 3.5.3. On débute en effet pour ce projet avec le groupe du verbe, puis le groupe du sujet, et finalement le groupe du complément. Chacun de ces trois groupes doit donc lui aussi fournir en sortie cette même information pour chaque phrase, soit la liste de mots, la liste de lemmes, la liste de classes grammaticales et d'étiquettes. La phrase finale se bâtit ensuite en joignant toutes ces informations. Les prochaines sections décrivent donc les algorithmes informatiques servant à bâtir chacun de ces trois groupes.

Mais nous débutons par discuter de méthodes communes utilisées par les algorithmes des trois groupes de la phrase. En premier lieu, une méthode concernant l'application du concept de « cooccurrences ». Ce concept vise à introduire un potentiel sémantique en encourageant le regroupement de mots qu'on retrouve souvent dans une même phrase dans le corpus de référence, tel que discuté précédemment à la Section 3.3. Comme les cooccurrences jouent un rôle au sein des trois groupes de la phrase (verbe, sujet et complément), il est pertinent d'en aborder l'utilisation en premier lieu (Section 4.10.1). Par la suite, on enchaine avec une brève discussion (Section 4.10.2) de la méthode utilisée pour sélectionner un mot au hasard parmi une liste accompagnée de fréquences d'apparition dans le corpus de référence.

4.10.1. Extraction des cooccurrences pour tous les mots de la phrase

Lors de la génération de phrases au hasard, on tente de regrouper les mots apparaissant souvent dans la même phrase, dans le corpus de référence, dans le but d'injecter un minimum de sens aux phrases. Pour y arriver, un vecteur de type « *String* », initialisé pour chaque phrase, emmagasine d'abord les lemmes de tous les noms communs, adjectifs, verbes et adverbes utilisés jusqu'à présent dans la phrase. À chaque fois qu'un mot est incorporé à une phrase, ce vecteur est mis à jour en y ajoutant le lemme du mot en question. L'extrait de code 4.8 montre la méthode très simple servant à ajouter un lemme au vecteur déjà existant.

La méthode reçoit donc en entrée la liste déjà existante de lemmes de la phrase, qui est vide au moment de débuter la phrase. La méthode reçoit aussi le nouveau lemme à ajouter à la liste existante. En sortie, cette méthode retourne la liste de lemmes mise à jour. Cette méthode est donc effectuée à chaque fois qu'un nouveau verbe, adjectif, nom ou adverbe est inséré à une phrase. Toutefois, l'algorithme permet de limiter la recherche de cooccurrences dans un environnement plus restreint, en se limitant à un nombre fixe de mots voisins à considérer, dans le but de favoriser les cooccurrences les plus pertinentes.

La Figure 4.42 illustre le fonctionnement de cette méthode pour une phrase fictive simple, étape par étape. À la première colonne, on voit l'ordre dans lequel les mots de la phrase sont générés aléatoirement. À la deuxième colonne, on voit croître la liste de lemmes (seulement des quatre types pertinents, soit verbes, adjectifs, noms et adverbes) à mesure qu'un mot se rajoute à la phrase. Pour chaque lemme, cette information se retrouve dans un objet de type « *OccurObjet* », tel que discuté à la Section 4.4. La méthode *TrouverOccur,* qui sert à extraire les cooccurrences, est illustrée à l'extrait de code 4.9. On y constate que l'objet *tableauOccur* est une table de hachage dont la clé est un lemme (*String*) et dont la valeur est un objet de type *OccurObjet*, discuté à la Section 4.4. La table de hachage *tableauOccur* contient donc toute l'information en lien avec toutes les cooccurrences de tous les lemmes d'intérêt du corpus de référence.

Extrait de code 4.8 : Méthode pour l'ajout des mots (noms communs, adjectifs, adverbes et verbes) de la phrase, pour en extraire plus tard les cooccurrences

```
public static String[] AjoutCooc(String[] originale, String ajout){

int taille=originale.length;
String [] finale = Arrays.copyOf(originale, taille+1);
finale[taille]=ajout;
return finale;
}
```

Mots de la phrases ajoutés dans l'ordre	Vecteur de cooccurences (lemmes)	Cooccurrences par classes grammaticales			
		Verbes	Adjectifs	Noms	Adverbes
se **dirige**	**diriger**	**aller (24), voir (13)**		**voiture (6) montagne (9)**	**lentement (8), précisément (3)**
se dirige **lentement**	diriger, **lentement**	aller (24), voir (13), **diriger (8) marcher (12), cuire (1)**	**beau (5), fini (3)**	voiture (6), montagne (9), **homme (3), machine (7), pied (1)**	lentement (8), précisément (3)
voiture se dirige lentement	diriger, lentement, **voiture**	aller (24), voir (13), diriger (8**+3**) marcher (12), cuire (1), **conduire (13), stationner (5)**	beau (5**+4**) fini (3), **cher (6), luxueux (2)**	voiture (6), montagne (9), homme (3), machine (7), pied (1)	lentement (8), précisément (3)
Cette **belle** voiture se dirige lentement	diriger, lentement, voiture, **beau**	aller (24), voir (13), diriger (11) marcher (12), cuire (1), conduire (13) stationner (5), **être (13), regarder (3)**	beau (9), fini (3), cher (6), luxueux (2), **gentil (4)**	voiture (6**+3**) montagne (9) homme (3), machine (7), pied (1)	lentement (8**+1**), précisément (3)
Cette belle voiture se dirige lentement vers les **montagnes**	diriger, lentement, voiture, beau, **montagne**	aller (24), voir (13), diriger (11**+2**) marcher (12), cuire (1), conduire (13) stationner (5), être (13), regarder (3)	beau (9), fini (3), cher (6), luxueux (2), gentil (4) **haut (4)**	voiture (9), montagne (9), homme (3), machine (7), pied (1), **neige (10), escalade (2)**	lentement (9), précisément (3), **difficilement (3)**

Figure 4.42 : Illustration du fonctionnement de la méthode « *AjoutCooc* » appliquée au moment de la création de la phrase « Cette belle voiture se dirige lentement vers les montagnes ». À mesure qu'un mot est intégré à la phrase, le vecteur de cooccurrences (2e colonne) croît. Les 4 tables de hachage (verbes, adjectifs, noms et adverbes) pour les cooccurrences croissent aussi, en ajoutant pour chaque nouveau mot de la phrase sa propre liste de cooccurrences. En rouge, les ajouts correspondant à chaque ligne. Entre parenthèses, les fréquences d'apparition des cooccurrences dans le corpus de référence

La méthode *TrouverOccur* retourne donc, pour un lemme fourni en entrée, toutes ses cooccurrences, groupées selon les quatre classes habituelles, soit les noms, adjectifs, verbes et adverbes. Il suffit ensuite d'effectuer cette méthode pour chaque mot de la phrase appartenant à ces classes, pour obtenir la liste totale de toutes les cooccurrences. C'est donc parmi celles-ci que les mots subséquents de la phrase sont sélectionnés au hasard. Grâce aux méthodes de type « *Get* » de la classe *OccurObjet* (extrait de code 4.4), on peut alors extraire la table de hachage contenant les lemmes de la classe qui nous intéresse, soit les verbes, adverbes, noms communs ou adjectifs.

Aux quatre colonnes suivantes se greffent ensuite toutes les cooccurrences associées à chaque lemme incluant leur fréquence d'apparition, selon les informations recueillies au cours de l'analyse du corpus de référence. Par exemple, le verbe « diriger » est associé dans le corpus aux verbes « aller » et « voir », aux noms « voiture » et « montagne », et aux adverbes « lentement » et « précisément ». À chaque nouvelle ligne, on ajoute aux lemmes déjà présents ceux correspondant au nouveau lemme ajouté au vecteur de cooccurrences. Quand un mot est cooccurrent avec plus d'un mot déjà inclus dans la phrase, on additionne les fréquences (voir par exemple la somme « 3+8 » pour le lemme « diriger » lors de l'ajout du lemme « voiture »).

La sélection d'un nouveau mot au hasard se fait donc parmi la liste de cooccurrences fournies dans les quatre tables de hachage de la Figure 4.42. Par exemple, l'adverbe « lentement » a été choisi car il représentait l'une des deux options d'adverbes cooccurrents au lemme « diriger ». De la même façon, le nom commun « voiture » a pu être sélectionné, car il faisait partie de la table de hachage pour les noms communs, et ainsi de suite pour le restant de la phrase. Il est à noter que la séquence dans laquelle les mots sont générés a une influence sur les mots sélectionnés, puisque les tailles des tables de hachage croissent à chaque nouveau lemme ajouté. Plus la phrase s'allonge, plus le choix de nouveaux mots potentiels augmente.

Extrait de code 4.9 : Méthode pour extraire toutes les cooccurrences d'un lemme en particulier fourni en entrée. En sortie, les cooccurrences sont fournies dans 4 tables de hachage, correspondant aux 4 classes de mots d'intérêt (noms, verbes, adverbes et adjectifs)

```
public OccurObjet TrouverCooccur(String lemme){

HashMap<String,Integer> noms;
HashMap<String,Integer> verbes;
HashMap<String,Integer> adverbes;
HashMap<String,Integer> adjectifs;
OccurObjet classes = new OccurObjet();

boolean existe=tableauOccur.containsKey(lemme);
if(existe){
    classes   = tableauOccur.get(lemme);
    noms      = classes.GetNoms();
    verbes    = classes.GetVerbes();
    adverbes  = classes.GetAdverbes();
    adjectifs = classes.GetAdjectifs();
}
return classes;
}
```

4.10.2. Méthode pour la sélection des mots au hasard

La génération de phrases aléatoires implique forcément la sélection de mots au hasard. La méthode pour sélectionner les mots aléatoirement est illustrée à l'extrait de code 4.10. En entrée, lors d'un appel de la fonction *motHasard*, on fournit une table de hachage. Les clés de cette table contiennent les lemmes des mots à choisir au hasard. Ces lemmes appartiennent tous à la même classe grammaticale (verbe, adverbe, nom commun, ou adjectif). Les valeurs de la table contiennent les fréquences d'apparition absolues de ces lemmes, donc des valeurs entières. Ces fréquences d'apparition sont soit celles dans le corpus de référence, ou celles parmi les cooccurrences (Section 4.10.1), dépendamment du contexte où la fonction *motHasard* a été appelée.

Si la table de hachage fournie en entrée est vide, la fonction retourne un mot (*String*) vide. La méthode ayant appelé la fonction *motHasard* s'occupera de bien traiter cette éventualité. Si la table de hachage ne contient qu'un mot, c'est évidemment ce mot qui est choisi. Cependant, si la table de hachage contient plus d'un mot, on doit faire intervenir le hasard.

Le Tableau 4.21 donne un exemple fictif de ce que cette table de hachage pourrait contenir, dans le cas des cooccurrences de type nom commun du nom « montre ». À la lecture du Tableau 4.21, on en conclut que trois phrases du corpus contiennent à la fois le lemme « aiguille » et le lemme « montre », deux phrases contiennent à la fois le lemme « poignet » et le lemme « montre », et ainsi de suite.

Tableau 4.21 : Exemple fictif du contenu d'une table de hachage envoyée en entrée à la fonction « *motHasard* ». Ici, c'est une table de noms communs correspondant au lemme « montre »

Clés de la table de hachage (*String*)	Valeurs de la table de hachage (entiers)
aiguille	3
poignet	2
heure	8
cadran	1
voleur	2

En entrée pour la méthode *motHasard*, on doit aussi fournir une variable « *methode* » de type entier et un paramètre « *param* » de type double. Deux options sont offertes pour sélectionner un mot au hasard. Dans la première, correspondant à « *methode*=0 », à chaque mot (clé) de la table, on associe un poids ou probabilité relative correspondant à sa fréquence d'apparition, donc à la valeur de la table de hachage. Ainsi, pour cette première option, la probabilité que le mot d'indice i soit choisi se calcule avec l'Équation 4.6.

$$p_i = \frac{h_i}{\sum_{j=1}^{j=n} h_j} \tag{4.6}$$

À l'Équation 4.6, p_i est la probabilité entre 0 et 1 du mot d'indice i, h_i est la valeur de table de hachage du mot d'indice i et $h_j$est la valeur de la table du mot d'indice j, où l'indice j va de 1 au nombre d'entrées dans la table de hachage.

Si la variable « *methode* » en entrée n'est pas égale à zéro, alors la probabilité de chaque mot devient fonction de la fréquence de ce mot élevée à un exposant « *param* », fourni lui aussi en entrée. Ainsi, pour cette deuxième option, la probabilité que le mot d'indice i soit choisi se calcule avec l'Équation 4.7.

$$p_i = \frac{{h_i}^{param}}{\sum_{j=1}^{j=n} {h_j}^{param}} \tag{4.7}$$

La Figure 4.43 illustre les probabilités pour chaque mot du Tableau 4.21 selon ces deux options. Pour la deuxième option, un exposant de valeur « 2.0 » (*param*) a été utilisé. À l'analyse de la Figure 4.43, on constate qu'élever les probabilités à un exposant supérieur à « 1 » fait augmenter la probabilité relative du mot qui était déjà le plus probable (ici, « heure »), en diminuant, en contrepartie, les probabilités des mots les moins probables. Cette option et cette valeur précise de la variable « *param* » (« 2.0 ») ont été appliquées pour ce projet, dans le but de mettre davantage en valeur les mots les plus probables. Dans le contexte des cooccurrences, cela a pour but de favoriser l'usage des mots qu'on retrouve le plus souvent ensemble dans une même phrase.

Extrait de code 4.10 : Sélection d'un mot au hasard parmi une table de hachage comportant les lemmes et leurs fréquences

```
public static String MotHasard(Hashtable<String,Integer> table,
                               int methode, double param){

String motChoisi;
if(methode==-1) methode=0;  if(methode==0 ) param=1.0;
if(param==-1.0) param=1.0;
int indice0,indice1,indicePrecedent; int stepIndice=1; int taille=table.size();
String [] lemmes = new String [taille];
double [] poids  = new double [taille];
double [] cumul  = new double [taille];
double total=0; double valeur0, valeur1;

Set<String> keys = table.keySet();
int tailleKey=keys.size();

if(table.size()==0){
    motChoisi="";return motChoisi;
}
if(table.size()==1){
   motChoisi=keys.toArray()[0].toString(); // motChoisi = seul mot de la table
}
else{ // Ici, plus d'un mot, donc y aller avec la procédure au hasard

// Définir les poids - si méthode=0, poids=fréquence inaltérée
for(int i=0;i<tailleKey;i++){ // passer à travers tous les mots de la table
    String lemme=keys.toArray()[i].toString();
    int freq=table.get(lemme);
    lemmes[i]=lemme;
    poids[i]=Math.pow((double)freq,param);
    total=total+poids[i];
    cumul[i]=total;
} // fin du for i keys size

// Déterminer un chiffre au hasard
double hasard = Math.random()*total;

// Déterminer point initial et direction itération
indice0=(int)((hasard/total)*taille);
valeur0=cumul[indice0]-hasard;
if(valeur0->0)stepIndice=-1;
indice1=indice0;

// Itération pour choisir le mot
do{
    indicePrecedent=indice1;
    indice1=indice1+stepIndice;
    if(indice1>=0&indice1<taille)valeur1=cumul[indice1]-hasard;
     else valeur1=valeur0;
} while((valeur0*valeur1)>0&indice1>=0&indice1<taille);

if(stepIndice==-1) motChoisi=lemmes[indicePrecedent];
else               motChoisi=lemmes[indice1];

return motChoisi;
}
```

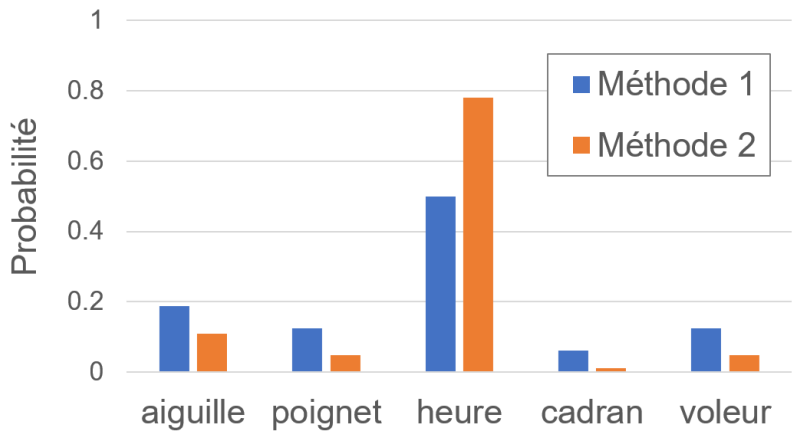


Figure 4.43 : Probabilité de sélection des mots du Tableau 4.21, selon les deux options de la fonction « *motHasard* ». Pour la deuxième option, un exposant (*param*) de 2.0 a été utilisé

4.10.3. Création de l'objet du groupe du verbe

La toute première étape pour construire le groupe du verbe consiste à sélectionner au hasard le verbe à utiliser. On sélectionne le verbe parmi les verbes répertoriés dans le corpus de référence, et selon les fréquences observées. Autrement dit, plus un verbe apparait souvent dans le corpus, plus souvent il est sélectionné pour la génération de phrases au hasard. La méthode utilisée pour choisir le verbe au hasard est illustrée à l'extrait de code 4.11. En entrée de cette méthode, on spécifie d'abord si on sélectionne le verbe parmi la liste de verbes « ambigus » ou « non ambigus ». Il faut se rappeler qu'au moment de la lemmatisation, certains mots peuvent demeurer ambigus, autrement dit, on ne peut déterminer à coup sûr quelle est leur classe grammaticale. Par exemple, si le nom commun « dame » est utilisé dans le corpus de référence, le verbe « damer », dont « dame » est une forme conjuguée, se retrouvera dans la liste des verbes « ambigus », à moins que le verbe « damer » soit utilisé ailleurs dans le texte, sous d'autres formes non ambiguës, par exemple « damerons ». On a donc ici l'option, par la variable booléenne « *ambigu* » fournie en entrée, de limiter l'usage de formes verbales ambiguës, comme « dame » dans l'exemple fourni ici. Si on choisit d'inclure les formes ambiguës, on sélectionne le verbe dans la table de hachage « *verbesA* ». Sinon, on le sélectionne dans la table de hachage « *verbesNA* ».

La méthode requiert aussi en entrée un vecteur *cooc* de type *String* contenant la liste de lemmes des mots (verbes, adjectifs, noms, adverbes) contenus dans la phrase jusqu'à présent. Cette liste de lemmes permet d'établir la liste de cooccurrences parmi lesquelles choisir le verbe. La méthode *VerbeHasard* offre donc deux options. Si le vecteur *cooc* est vide, on sélectionne au hasard le verbe parmi tous les verbes du corpus de référence, en fonction de leurs fréquences d'apparition. Si au contraire le vecteur *cooc* n'est pas vide, on choisit le verbe uniquement parmi les cooccurrences des mots intégrés à la phrase jusqu'à présent (Section 4.10.1). Mais au moment de créer une nouvelle phrase, comme on débute par le verbe, le vecteur *cooc* est forcément vide. On ne peut donc pas se servir de cooccurrences pour le choix du verbe principal du groupe du verbe.

Extrait de code 4.11 : Sélection d'un verbe au hasard

```
public String VerbeHasard(boolean ambigu, String [] cooc){

String infinitif;
OccurObjet classes;
Hashtable<String,Integer> verbeCooc;

int coocTaille=cooc.length;
    if (cooc.length==0){ // Pas de cooccurrences fournies
        if(ambigu) infinitif=Ut.MotHasard(verbesNA, 1, 1.0);
        else       infinitif=Ut.MotHasard(verbesA, 1, 1.0);}
    else{               // Cooccurrences fournies
        Hashtable<String,Integer>[] arrayTables;
        arrayTables = new Hashtable[cooc.length];
        for(int i=0;i<coocTaille;i++){
            classes=TrouverCooccur(cooc[i]);
            arrayTables[i]=classes.GetVerbes();
        } // fin du for i
        verbeCooc=Ut.CombineFreq(arrayTables);
        verbeCooc=Ut.CoocMemePhrase(verbeCooc, cooc);
        infinitif=Ut.MotHasard(verbeCooc, 1, 1.0);
    } // fin du else cooccurrences fournies

return infinitif;
}
```

Lorsque le vecteur *cooc* n'est pas vide, on extrait donc les cooccurrences de tous les lemmes qu'il contient grâce à la méthode *TrouverOccur* (extrait de code 4.9). On combine ensuite tous ces lemmes en une liste unique, avec un appel à la méthode « *CombineFreq* » (code non illustré ici). Cette méthode s'assure que chaque lemme cooccurrent n'apparaît qu'une seule fois et qu'il combine toutes les fréquences d'apparition associées à chaque lemme. Finalement, on s'assure, avec la méthode *CoocMemePhrase* (code non illustré ici), que les mots déjà inclus dans la phrase n'apparaissent pas eux-mêmes dans la liste de cooccurrences. Dans le cas contraire, on risquerait de générer au hasard des phrases contenant le même mot plus d'une fois (par exemple, « Le garçon regarde le garçon »). Finalement, on détermine le verbe au hasard grâce à la méthode « *MotHasard* » (extrait de code 4.10) discutée à la section précédente.

Une fois le verbe lui-même choisi, on sélectionne le temps de verbe au hasard. À l'Étape 1, les fréquences des temps de verbe avaient été calculées pour le corpus de référence. Deux séries de valeurs avaient en fait été calculées : une considérant les formes ambiguës (*ratioTempsA*), l'autre ne considérant que les formes de verbes non ambiguës (*ratioTempsNA*). La sélection du temps de verbe pour l'Étape 2 s'effectue donc, au choix, selon l'une de ces deux séries de valeurs. Il est toutefois à noter qu'une fonction (non illustrée ici) a été programmée afin d'assurer, au besoin, que chacun des temps de verbes possibles soit associé à une fréquence minimale. Ainsi, si par exemple le subjonctif passé n'est jamais utilisé dans le corpus de référence, on peut quand même s'assurer qu'il apparaisse dans les phrases générées aléatoirement à l'Étape 2.

À partir de la série de valeurs choisies (*ratioTempsA*, *ratioTempsNA*, ou une version modifiée de celles-ci), on crée une table de hachage contenant tous ces ratios de temps. Une fois la table de hachage bâtie, on détermine le temps au hasard, en appelant la fonction « *DoubleHasard* », qui accepte en entrée la table de hachage, un paramètre indiquant une méthode à utiliser (entier) et un exposant (double). La fonction « *DoubleHasard* » (non illustrée ici) est très semblable en fonctionnalité à la fonction « *motHasard* » fournie à l'extrait de code 4.10. La différence est que la fonction « *DoubleHasard* » accepte en entrée une table de hachage dont les clés sont des entiers

(codes pour temps de verbes) et les valeurs sont des variables de type « double » (les ratios des temps de verbes). La sélection du temps de verbe se fait donc avec l'extrait de code 4.12.

Une fois le temps du verbe déterminé, on doit déterminer aléatoirement la personne du verbe. Le processus est en tout point semblable à la sélection du temps du verbe. À l'Étape 1, les fréquences des personnes de verbe avaient été calculées pour le corpus de référence. Deux séries de valeurs avaient en fait été calculées : une considérant les formes ambiguës (*ratioPersonnesA*), l'autre ne considérant que les formes de verbes non ambiguës (*ratioPersonnesNA*). La sélection de la personne pour l'Étape 2 s'effectue donc, au choix, selon l'une de ces deux séries de valeurs. Il est toutefois à noter qu'une fonction (non illustrée ici) a été programmée afin d'assurer, au besoin, que chacune des personnes soit associée à une fréquence minimale. Ainsi, si par exemple la première personne du pluriel n'est jamais ou très peu utilisée dans le corpus de référence, on peut quand même s'assurer qu'elle apparaisse dans les phrases générées aléatoirement à l'Étape 2.

Pour sélectionner la personne du verbe, il faut toutefois tenir compte du fait qu'à l'impératif, seules trois personnes sont possibles, soit la deuxième du singulier, et les première et deuxième personnes du pluriel. Pour l'impératif, il faut donc limiter le choix à ces trois personnes. Par simplicité, les personnes de l'impératif ne sont pas basées sur les fréquences d'apparition dans le corpus de référence, mais plutôt complètement au hasard, avec probabilités égales pour les trois cas. L'extrait de code 4.13 illustre la sélection de la personne. Il est à noter que l'impératif correspond au code « 7 », et que les personnes sont codées de 1 à 6, selon le Tableau 3.5.

Extrait de code 4.12 : Sélection d'un temps de verbe au hasard

```
for(int i=0;i<21;i++){
   if(ambigu) table.put(i, ratioTempsA[i]);
   else       table.put(i, ratioTempsNA[i]);}
temps=Ut.DoubleHasard(table, 1, 1.0);
```

Extrait de code 4.13 : Sélection de la personne du verbe au hasard

```
if(temps!=7){ // pas à l'impératif
    for(int i=0;i<7;i++){
       if(ambigu) tableP.put(i, ratioPersonnesA [i]);
       else       tableP.put(i, ratioPersonnesNA[i]);}
       personne=Ut.DoubleHasard(tableP, 1, 1.0);
    }
else{ // impératif - une des trois personnes au hasard, poids égaux
    personne=5;
    double hasardImp=Math.random();
    if(hasardImp<0.33) personne=2;
    else{
        if (hasardImp<0.66) personne=4;
    }
}
```

Finalement, une vérification additionnelle s'effectue, dans le cas où le verbe sélectionné au hasard est impersonnel, par exemple, « falloir ». Comme de tels verbes ne se conjuguent qu'à la troisième personne du singulier, l'algorithme impose alors cette personne, faisant fi de celle déterminée au hasard à l'extrait de code 4.13.

Les paramètres du verbe sélectionné au hasard sont ensuite extraits, à partir de la table de hachage « *infinitifH* » à l'aide de méthodes de type « *Get* » appliquées à l'objet. Cet objet est de type « *VerbeInf* », tel que décrit à l'extrait de code 4.2. Les paramètres en question sont ceux listés dans le Bescherelle (2012) pour chaque infinitif. On y retrouve les possibilités que le verbe soit utilisé aux modes « transitif direct », « transitif indirect », « intransitif », ou « pronominal ». Dans le cas où un verbe peut être utilisé avec un complément d'objet indirect, des prépositions appropriées sont aussi fournies. Si la forme est pronominale, le groupe du verbe inclut le pronom réfléchi.

Quand plusieurs modes sont possibles pour un verbe donné parmi « transitif direct », « transitif indirect » ou « intransitif », on détermine ce mode au hasard, avec une probabilité égale pour chacun. Ce mode n'affecte en rien la formation du groupe du verbe, tel que défini pour ce projet. Cette information est plutôt utile pour bien y arrimer le groupe du complément à une étape ultérieure (Section 4.10.5). La préposition à utiliser, le cas échéant, est aussi déterminée au hasard, si plusieurs options existent. L'information sur le mode et la préposition choisis est donc fournie en sortie de la méthode du groupe du verbe.

On doit ensuite déterminer au hasard si le verbe sera utilisé dans une forme modale, donc précédé d'un verbe modal. Les verbes modaux permettent des formes telles que « je dois manger » ou « je veux manger ». Pour déterminer si le verbe sera de la forme modale, on effectue un calcul de hasard simple, basé sur une valeur seuil fournie comme paramètre et modifiable au gré de l'utilisateur. Si l'option modale est choisie, il faut alors sélectionner le verbe modal qui sera utilisé parmi ceux du Tableau 3.20. On procède en assignant arbitrairement à chaque verbe modal un « poids ». On crée ensuite une table de hachage contenant chaque verbe modal comme clé (*String*) et son poids (entier) comme valeur. On peut ensuite utiliser la méthode « *motHasard* » décrite précédemment à l'extrait de code 4.10. La méthode pour choisir au hasard le verbe modal est illustrée à l'extrait de code 4.14. De la même façon que le modal, on détermine au hasard, toujours selon un paramètre seuil arbitraire, si le verbe sera sous la forme affirmative (« je mange ») ou négative (« je ne mange pas »).

On procède ensuite à la conjugaison du verbe, qui dépend de l'infinitif, du temps et de la personne. Deux cas sont considérés, selon que le verbe est conjugué ou non à un temps composé. Si le verbe est à un temps composé (participe passé, plus-que-parfait, etc.), c'est l'auxiliaire « avoir » ou « être » qui doit être conjugué. La forme au participe passé du verbe sélectionné y est ensuite juxtaposée. Si au contraire le temps sélectionné n'est pas composé, on conjugue tout simplement le verbe sélectionné tel quel. L'algorithme doit donc considérer ces deux cas séparément.

Mais il faut aussi considérer le cas où le verbe est conjugué sous la forme modale. Dans le cas d'un temps composé, on utilise le participe passé du verbe modal, qui est ensuite suivi de l'infinitif du verbe sélectionné au départ. Dans le cas d'un temps non composé, c'est le modal qui doit être conjugué, suivi encore une fois de l'infinitif du verbe sélectionné au départ. Le Tableau 4.22 illustre les différents cas à considérer pour la conjugaison du verbe. L'extrait de code 4.15 illustre comment les quatre cas du Tableau 4.22 sont traités.

Extrait de code 4.14 : Sélection du verbe modal

```
public String ChoisirModal(){

String choix;
String [] modaux = new String [11];
Integer[] poids  = new Integer[11];

modaux[0] ="aimer";    poids[0]=1;
modaux[1] ="croire";   poids[1]=1;
modaux[2] ="devoir";   poids[2]=10;
modaux[3] ="espérer";  poids[3]=1;
modaux[4] ="paraître"; poids[4]=1;
modaux[5] ="penser";   poids[5]=1;
modaux[6] ="pouvoir";  poids[6]=10;
modaux[7] ="savoir";   poids[7]=1;
modaux[8] ="sembler";  poids[8]=1;
modaux[9] ="vouloir";  poids[9]=10;
modaux[10]="aller";    poids[10]=10;

HashMap<String,Integer> table = new HashMap<>();
for(int i=0;i<11;i++){
    table.put(modaux[i], poids[i]);
}
choix=Ut.MotHasard(table, 0,1.0);
return choix;
}
```

Tableau 4.22 : Les cas à considérer pour la conjugaison selon que le temps de verbe est composé ou non, et que la forme modale est utilisée ou non, en utilisant le verbe « manger » et le modal « devoir ». Les formes conjuguées sont surlignées en rouge gras, les participes passés sont surlignés en bleu, et les infinitifs sont surlignés en vert

Temps de verbe	Mode du verbe	Exemple
Non composé	Non modal	« Je **mange** »
	Modal	« Je **dois** manger »
Composé	Non modal	« J'**ai** mangé »
	Modal	« J'**ai** dû manger »

Extrait de code 4.15 : Considération des quatre cas possibles pour la conjugaison, selon que le temps est composé ou non, et que le verbe est sous la forme modale ou non

```
if(temps<12){ // Temps non composés
    if(modal){ // Si modal, verbe conjugué est le verbe modal pour non composé
      verbeC=Conjugue(verbeModal, temps, personne);
    }
    else verbeC=Conjugue(infinitif, temps, personne);
} // fin du if temps non composés
else{ // Temps composés - besoin auxiliaire conjugué + pp
    if(auxEtre&auxAvoir){ // se conjugue avec être ou avoir - choisir au hasard
        if(Math.random()>0.5) auxAvoir=false; else auxEtre=false;
    }
    if(auxEtre)verbeAux="être"; else verbeAux="avoir";
    verbeC=Conjugue(verbeAux, temps-11, personne); // Conjuguer l'auxiliaire

    // Choix du participe passé
    if(auxAvoir) personnePP=1; // Conjugué avec avoir - pp ne s'accorde pas
    else{                      // Conjugué avec être  - pp doit s'accorder
      if(Math.random()>0.5) genre=0; else genre=1; // Genre déterminé au hasard
      personnePP=genre*2+nombre+1; // valeurs de 1 à 4 pour pp selon genre/nombre
    }

    if(modal){ // Si modal, le participe passé est celui du verbe modal
      participePasse=Conjugue(verbeModal, 10, personnePP);
      verbePP=verbeModal;//
    }
   else participePasse=Conjugue(infinitif, 10, personnePP);
   verbePP=infinitif;// Conjuguer le pp
} // fin du else temps composé
```

L'extrait de code 4.15 indique qu'on doit aussi tenir compte, dans les cas de temps composés, de l'auxiliaire (« être » ou « avoir ») qui doit être utilisé avec le verbe. L'auxiliaire à utiliser pour chaque verbe est spécifié dans l'objet « *verbeInf* » pour chaque infinitif. Il est à noter que certains verbes s'emploient avec l'un ou l'autre. Le hasard est donc appliqué pour déterminer celui à utiliser.

L'extrait de code 4.15 incorpore des appels à la méthode « *Conjugue* », qui est celle qui procède à la conjugaison en tant que tel. Cette méthode, comme on peut s'y attendre, accepte en entrée le verbe à conjuguer, le temps, et la personne. Cette méthode « *Conjugue* » effectue exactement ce qu'un professeur du primaire exigerait de ses élèves, par exemple « conjugue-moi le verbe *manger* au passé-composé, à la première personne du pluriel ». Cette méthode doit d'abord extraire le modèle de conjugaison du verbe en entrée, selon les quelques dizaines de modèles du Bescherelle. Une fois le modèle et sa colonne attitrée déterminés, on doit trouver la ligne de la banque de données de conjugaison se rapportant au temps de verbe et à la personne du verbe fournis en entrée, tel qu'on le voit au Tableau 4.3 des conjugaisons. Ensuite, on bâtit la forme conjuguée en juxtaposant le radical du verbe à la terminaison. Il faut toutefois tenir compte du fait que certains modèles de conjugaison sont applicables à des radicaux donnant une certaine flexibilité. Par exemple, tel que mentionné précédemment, le Modèle 9 est caractérisé par des terminaisons d'infinitif de type « e*er ». Pour le verbe « peser », le caractère « * » est associé à la lettre « s ». Tandis que pour le verbe « amener » qui se conjugue selon le même modèle, le caractère « * » correspond à la lettre « n ». L'extrait de code 4.16 illustre la méthode « *Conjugue* » dans son entièreté.

Extrait de code 4.16 : Méthode pour conjuguer un verbe donné à un temps et une personne donnés

```
public String Conjugue(String verbe, int temps, int personne){

String radical, terminInf, replace, terminaison;
String verbeC="!";
String ligne[];
int modele, colonne, cellule, enlever, index, position;
boolean starVerif;
boolean arret=false;

while(!arret){
    modele=TrouveModele(verbe);
    colonne=TrouveColonneModele(modele);
    cellule=Ut.ColonneTab(temps, personne);
    if(cellule==-1){temps=0;cellule=1;}
    ligne=verbeTab[colonne];
    enlever=Integer.parseInt(ligne[4]);
    radical=verbe.substring(0,verbe.length()-enlever);
    terminInf=ligne[3];
    terminaison=ligne[cellule];
    starVerif=terminInf.contains("*"); // Quand modèle inclut lettre *

    if(starVerif){
        index=terminInf.indexOf("*");
        // Test pour deux consonnes pour le *
        String Pos4=verbe.substring(verbe.length()-4,verbe.length()-3);
        boolean testPos4=(Pos4.equals("e")|Pos4.equals("é"));
        int deltaPos=0;
        if(!testPos4){
            index--;
            deltaPos=1;
            radical=radical.substring(0,radical.length()-1);
        }
        position=verbe.length()-enlever+index;
        replace=verbe.substring(position, position+1+deltaPos);
        terminaison=terminaison.replace("*",replace);
    }
    verbeC=radical+terminaison;
    verbeC=verbeC.replace("$","");
    if(temps==0) verbeC=verbe;
    arret=true;
    if(Parametres.etapePhrase&&verbeC.contains("!")){ // Verbe non conjugable
        arret=false;
        String[] coocVide = new String[0];
        verbe=VerbeHasard(false, coocVide);
    } // Fin du if verbe non conjugable quand on bâtit des phrases
}  // Fin du while
return verbeC;
}
```

Il reste finalement à déterminer si des adverbes seront intégrés au groupe du verbe. Trois options sont possibles, tel que mentionné à la Section 3.5.3.1 : aucun adverbe, un adverbe de caractérisation, ou un adverbe d'intensité suivi d'un adverbe de caractérisation. Pour déterminer laquelle des trois options doit être appliquée, deux paramètres arbitraires sont utilisés. Le premier (*probAdverbe*) détermine la probabilité qu'on utilise ou non des adverbes, et si on utilise des adverbes, l'autre paramètre (*probIntensite*) détermine si on utilise un adverbe d'intensité.

L'extrait de code 4.17 montre comment ces trois options sont déterminées. Cet extrait inclut l'ajout des adverbes sélectionnés au vecteur de mots *coocPlus*, qui contient tous les mots de la phrase à date de type nom, adjectif, verbe et adverbe, dans le but d'influencer le choix des mots au hasard selon leurs cooccurrences. De plus, l'extrait de code 4.17 inclut un appel à la méthode *ChoisirAdverbe* (non illustrée ici), qui reçoit en entrée la liste de mots *coocPlus*, ainsi que le type d'adverbe qu'on recherche. Le code « 42 », tel que décrit au Tableau 3.2, correspond à des adverbes de caractérisation (« lentement »), tandis que le code « 41 » correspond à des adverbes d'intensité (« très »).

À ce stade, tous les paramètres servant à bâtir le groupe du verbe ont été déterminés. Il ne reste plus qu'à mettre les mots et lemmes bout à bout. Au Chapitre 3 (Section 3.5.3.1.1) on a fourni huit possibilités de séquences de mots pour le groupe du verbe. Celui-ci est donc bâti en suivant l'un de ces huit modèles, en fonction des conditions qui leur sont associées, grâce à des énoncés de type « *IF* » et « *ELSE* ».

Pour chaque forme, on détermine en premier lieu le nombre de mots (*nMotsGV*) que le groupe du verbe doit inclure. De façon générale, ce nombre de mots dépend du fait que la phrase puisse être à la forme affirmative ou négative, qu'elle soit sous la forme pronominale ou non, que le verbe soit composé ou non, et qu'on y incorpore ou non des adverbes de caractérisation et d'intensité. Une fois le nombre de mots déterminé, on initialise les vecteurs du groupe du verbe devant en contenir les mots (*motGV* – vecteur de type *String*), les lemmes (*lemmeGV* – vecteur de type *String*), les classes grammaticales (*POSGV* – vecteur de type entier) et les étiquettes morpho-syntaxiques (*POSdGV* – vecteur de type *String*). L'extrait de code 4.18 donne l'exemple de la Forme 8 correspondant aux verbes qui ne sont ni à la forme impérative, ni à la forme modale, pour des temps non composés. Des énoncés semblables sont générés pour les sept autres formes.

Extrait de code 4.17 : Code pour déterminer la présence d'adverbes et les sélectionner le cas échéant

```
// Déterminer si adverbe, puis choisir adverbe
adverbe=""; adverbeIntensite="";
if((Math.random()<probAdverbe)){ // début du if adverbe
    // Choisir l'adverbe
    adverbe=ChoisirAdverbe(coocPlus, 42);
    // Ajouter verbe à l'array de String de coocurrences
    coocPlus=Ut.AjoutCooc(coocPlus, adverbe);
    if((Math.random()<probIntensite)){
      String[] coocVide = new String[0];
      adverbeIntensite=ChoisirAdverbe(coocVide, 41);
    }// fin du if probIntensite
    coocPlus=Ut.AjoutCooc(coocPlus, adverbeIntensite);

}
```

Extrait de code 4.18 : Préparation des variables du groupe du verbe, pour la Forme 8 : Ni impératif, ni modal – Temps non composé

```
nMotsGV=1+(2*negatif)+Ut.BoolToInt(pronominal)+Ut.BoolToInt(compose)+
Ut.BoolToInt(!adverbeIntensite.equals(""))+Ut.BoolToInt(!adverbe.equals(""))+Ut.B
oolToInt(testEn&pronominal);

motGV   = new String[nMotsGV];
lemmeGV = new String[nMotsGV];
POSdGV  = new String[nMotsGV];
POSGV   = new int   [nMotsGV];
```

Une fois les variables de l'extrait de code 4.18 créées, on y insère un à un chacun de leurs éléments, selon la séquence de mots fournie à la Section 3.5.3.1.1. Le détail de cette opération assez triviale mais plutôt longue, n'est pas illustré ici.

Finalement, une fois tous ces vecteurs formés, on doit préparer l'information complète qui doit être fournie en sortie du groupe du verbe. Cette information servira ensuite à bien arrimer les groupes du sujet et du complément, et d'ultimement bâtir la phrase complète. Comme une méthode ne peut fournir qu'un seul élément en sortie (commande *return*), on crée un objet (classe *ObjetGV*) contenant toute cette information, et c'est cet objet qui est retourné par la méthode *GroupeVerbe*. Cette classe est illustrée à l'extrait de code 4.19. Cette classe comprend un constructeur qui initialise les vecteurs selon la dimension requise. Elle comprend ensuite une méthode de type « *Set* » qui sert à assigner toutes les valeurs à toutes les variables de la classe (*SetGV*). Plusieurs des variables de la méthode *SetGV* serviront de paramètres d'entrée pour les méthodes servant à bâtir le groupe du sujet et le groupe du complément. Cette classe ne comprend pas de méthode de type « *Get* ». Les variables de cette classe publique seront donc extraites directement par des références du type *ObjetGV.nomdevariable* lorsque requises au sein des autres méthodes.

Extrait de code 4.19 : Classe « *ObjetGV* », fournissant l'objet en sortie de la méthode du groupe verbe

```
public class ObjetGV {

    int nMotsGV, tempsGV, genreGV, nombreGV, nCoocGV, personneGV, posPPGV;
    String texteGV, infinitifGV, prepositionGV, verbePPGV;
    String[] motsGV, lemmesGV, coocGV, POSdGV;
    int [] POSGV;
    boolean impersonnelGV, subjonctifGV, intransitifGV, attributifGV, imperatifGV,
    auxAvoirGV, negatifGV, CODmassifGV, finaleDeGV, modalGV, pronominalGV;

public ObjetGV(int nMots, int nCooc){ // CONSTRUCTEUR

   nMotsGV=nMots;
   nCoocGV=nCooc;
   motsGV   = new String[nMotsGV+1];
   lemmesGV = new String[nMotsGV+1];
   POSdGV   = new String[nMotsGV+1];
   coocGV   = new String[nCooc];
   POSGV    = new int   [nMotsGV+1];}

public void SetGV(int temps, int genre, int nombre, int personne,
      String texte, String infinitif, String preposition,
      String [] mots, String [] lemmes, String [] POSd, String [] cooc,
      int [] POS,
      boolean impersonnel, boolean subjonctif, boolean intransitif,
      boolean attributif, boolean imperatif, boolean auxAvoir,
      boolean negatif, boolean CODmassif, String verbePP, boolean modal,
      boolean pronominal, int posPP){
    texteGV=texte;
    tempsGV=temps; genreGV=genre; nombreGV=nombre;
    motsGV = mots; lemmesGV=lemmes; coocGV=cooc;; POSdGV=POSd;
    POSGV=POS;
    posPPGV=posPP; // position du participe passé, si doit changer plus tard
    impersonnelGV=impersonnel;
    subjonctifGV=subjonctif; intransitifGV=intransitif;
    attributifGV=attributif; imperatifGV=imperatif; auxAvoirGV=auxAvoir;
    negatifGV=negatif; CODmassifGV=CODmassif;
    personneGV=personne;
    prepositionGV=preposition;
    modalGV=modal;
    pronominalGV=pronominal;
    verbePPGV=verbePP; // verbe au participe passé pour temps composés
                       // (l'infinitif choisi, ou le modal choisi)
}

public void SetMotsSansTexte(String [] mots, String [] lemmes, int[] POS, String[]
POSd){
    motsGV=mots;
    lemming=lemmes;
    POSGV=POS;
    POSdGV=POSd;
    nMotsGV=motsGV.length;}
}
```

4.10.4. Création de l'objet du groupe du sujet

Le cas le plus simple du groupe du sujet (méthode *GroupeSujet*) correspond au cas où le groupe du verbe est à l'impératif. Dans ce cas, le groupe du sujet retourne tout simplement une série de vecteurs vides pour les mots, lemmes et classes grammaticales. Il va donc de soi que la première étape de la création du groupe du sujet consiste à vérifier si le groupe du verbe est à l'impératif. Un autre cas très simple est celui où le verbe du groupe du verbe est impersonnel. Dans un tel cas, le groupe du sujet doit se limiter à « il » (« il pleut »), sauf dans le cas du subjonctif (voir plus bas), où on inclura d'abord une locution du type « il faut que » devant le « il ».

La deuxième étape de la création du groupe du sujet consiste à vérifier si le groupe du verbe est au mode subjonctif (subjonctif présent, imparfait, passé ou plus-que-parfait). Dans un tel cas, il faut ajuster le groupe du sujet (tel que défini pour ce projet) en conséquence. Bien que plusieurs formes puissent engendrer le subjonctif, seules des formes simples impliquant le verbe « falloir » sont utilisées pour ce projet. Ainsi, lorsque le groupe du verbe est à la forme subjonctive, il faut déterminer le temps du verbe « falloir » à employer pour l'arrimer au groupe du verbe. Si le groupe verbe est au subjonctif présent ou imparfait, l'algorithme choisit au hasard le temps du verbe « falloir » parmi ceux-ci : présent, imparfait, passé simple, futur simple, passé-composé, plus-que-parfait, passé antérieur ou futur antérieur. Les probabilités de chaque temps sont fonction de leurs fréquences respectives dans le corpus de référence. Si le groupe du verbe est au subjonctif passé ou au plus-que-parfait, le verbe « falloir » est automatiquement conjugué à l'imparfait. L'extrait de code 4.20 effectue ces tests pour déterminer le temps à utiliser pour le verbe « falloir ».

Il faut ensuite déterminer aléatoirement si le verbe « falloir » sera utilisé à la forme affirmative ou négative. Un paramètre (« *probSubjNegatif* ») est déterminé arbitrairement pour y arriver. Une fois le temps du verbe falloir et le choix de la forme affirmative ou négative déterminés, on peut bâtir le groupe du verbe s'y rapportant. On utilise alors le même objet que celui utilisé pour le groupe du verbe de la phrase, tel que discuté à la Section 4.10.3. Il est à noter que la personne utilisée avec le verbe « falloir » est toujours la troisième du singulier (« il »). L'extrait de code 4.21 donne le détail de ces opérations. L'appel à la méthode « *GroupeVerbe* » se fait ici en imposant évidemment le verbe « falloir », en spécifiant le temps choisi au hasard pour le subjonctif à l'extrait de code 4.20, la personne (code=3 pour « il ») et le paramètre déterminant si on utilise la forme affirmative ou négative (« *subjNegatif* »). On envoie le paramètre « -1 » pour le « mode », puisqu'on permet au hasard l'usage du modal par exemple. Finalement, on envoie « *coocVide* » qui est un vecteur vide de type *String* pour les cooccurrences, puisqu'en imposant le verbe « falloir », on ne se soucie pas des cooccurrences. L'appel à la méthode « *GroupeVerbe* » génère donc en sortie l'information dont on a besoin pour inclure au groupe du sujet pour le cas subjonctif, soit la liste de mots, la liste de lemmes, la liste de classes grammaticales et la liste des étiquettes morpho-syntaxiques. On juxtaposera ces vecteurs aux autres vecteurs semblables qui seront générés pour le reste du groupe du sujet.

Une fois la question du subjonctif réglée, il faut poursuivre avec le reste du groupe du sujet qui suit la locution du type « il faut que », ou tout simplement débuter le groupe du sujet à partir de zéro, dans le cas non subjonctif. La toute première étape à ce stage est de déterminer si le groupe du sujet sera basé sur des noms (« *le garçon* mange ») ou des pronoms (« *il* mange »). L'emploi de noms pour le groupe du sujet ne se fait qu'à la troisième personne du singulier (code=3) et à la troisième personne du pluriel (code=6). Autrement, on impose l'utilisation de pronoms (« je », « tu », « nous », et « vous »). À la troisième personne, on détermine donc au hasard, selon un paramètre arbitraire « *probNom* » si le groupe du sujet sera basé sur des noms ou des pronoms. L'extrait de code 4.22 illustre cette opération. La méthode pour déterminer le groupe du sujet se sépare ensuite en deux sections, l'une pour les pronoms (Section 4.10.4.1), l'autre pour les noms communs (Section 4.10.4.2).

Extrait de code 4.20 : Déterminer le temps du verbe « falloir » pour le cas où le groupe du verbe est au subjonctif, pour inclusion au groupe du sujet

```
// Tester si subjonctif
if(subjonctif){
  if(temps==5|temps==6){ //GV au subjonctif présent ou passé
      // Possibilités de temps à exclure: 0, 5, 6, 7, 9, 10, 11, 16, 17, 20
      for(int i=0;i<21;i++){
        boolean
test=(i==0)|(i==5)|(i==6)|(i==7)|(i==9)|(i==10)|(i==11)|(i==16)|(i==17)|(i==20);
        if(test)tableTempsSubj.put(i, 0.0);
        else    tableTempsSubj.put(i, ratioTempsA[i]);} // fin du for i
      tempsSubjonctif=Ut.DoubleHasard(tableTempsSubj, 1, 1.0);
  } // fin du temps=5 ou 6
  if(temps==16|temps==17){ //GV au subjonctif présent ou passé
      tempsSubjonctif=2; // Imposer l'imparfait - temps 16 et 17 très rares
  } // fin du temps = 16 ou 17
```

Extrait de code 4.21 : Bâtir un objet de type « *ObjetGV* » pour la forme « il faut que », pour inclure au groupe du sujet

```
ObjetGV groupeVerbeSubjonctif;

groupeVerbeSubjonctif=GroupeVerbe("falloir", tempsSubjonctif, 3, subjNegatif, -1,
coocVide);

nMotsSubj=groupeVerbeSubjonctif.nMotsGV+2; // Rajouter 2 mots au GV: IL faut QUE...
motsSubj   = groupeVerbeSubjonctif.motsGV;
lemmesSubj = groupeVerbeSubjonctif.lemmesGV;
POSSubj    = groupeVerbeSubjonctif.POSGV;
POSdSubj   = groupeVerbeSubjonctif.POSdGV;
```

Extrait de code 4.22 : Déterminer au hasard si le groupe du sujet est basé sur des noms communs ou des pronoms

```
if(personne==3|personne==6) avecNom=(Math.random()<probNom);
else avecNom=false;
```

4.10.4.1. Création de l'objet du groupe du sujet avec des pronoms

Plusieurs options de groupe du sujet avec pronoms sont proposées pour ce projet. Celles-ci ont été décrites au Tableau 3.28, accompagnées de certaines de leurs contraintes, en ce qui a trait au nombre (singulier ou pluriel) et à la forme affirmative ou négative du groupe du verbe.

Mais tel que mentionné précédemment, si le groupe du verbe est à la première ou à la deuxième personne, non seulement l'option du pronom est la seule possible, un seul type de pronom est permis, le pronom-sujet (nominatif). On ne peut utiliser en effet que les pronoms « je », « tu », « nous » et « vous » dans de tels cas. La première étape visant à créer le groupe du sujet avec pronoms est donc de déterminer si le groupe du verbe est à la première personne ou à la deuxième personne (singulier ou pluriel). Dans un tel cas, la première option de pronoms du Tableau 3.28 (pronom-sujet), associée au code « 0 », est imposée.

Lorsque le groupe du verbe est à la troisième personne du singulier ou du pluriel, on peut utiliser n'importe lequel des huit types de pronoms du Tableau 3.28, dans la mesure où les contraintes de nombre et de type de phrase (affirmative ou négative) sont respectées. Pour déterminer lequel des huit types de pronoms est choisi, on assigne arbitrairement un poids à chacun. Des énoncés conditionnels (« IF ») sont ensuite exécutés pour éliminer les cas soumis à des contraintes. Par exemple, on n'utilise le pronom « tous » qu'au pluriel et le pronom « rien » qu'à la forme négative. On élimine les cas non permis en remplaçant leur poids par défaut par une valeur de zéro. De la même façon qu'on l'avait fait pour déterminer au hasard les temps et personnes de verbes au groupe du verbe, on utilise la fonction « *DoubleHasard* », à laquelle on envoie une table de hachage dont les clés sont les codes des huit types de pronoms (valeurs entières) et les valeurs sont les poids fournis arbitrairement (valeurs doubles). L'extrait de code 4.23 donne le détail des opérations d'assignations de poids et de la détermination au hasard du type de pronom choisi.

Extrait de code 4.23 : Déterminer au hasard l'un des huit types de pronoms du Tableau 3.28 à utiliser, après avoir assigné un poids (probabilité relative) à chacun

```
// Cas n'impliquant aucune contrainte
pronomPoids.put(0,5.0);pronomPoids.put(1,5.0);
pronomPoids.put(3,5.0);pronomPoids.put(4,5.0);

// Cas impliquant contraintes
if(plur&pos) pronomPoids.put(2,5.0);
if(sing&neg){pronomPoids.put(5,5.0); pronomPoids.put(7,25.0);}
if(!neg&(!(genre==1&sing))) pronomPoids.put(6,5.0);
if(!plur) pronomPoids.put(8,5.0);

// Choix du type de pronom
if(personne==3|personne==6)casPronom=Ut.DoubleHasard(pronomPoids, 1, 1.0);
else casPronom=0; // je tu nous vous si pas à la 3e pluriel ou singulier
```

On détermine ensuite le nombre de mots (*nMotsGS*) que le groupe du sujet doit inclure. De façon générale, ce nombre de mots est égal à un (le pronom lui-même). Cependant, il faut ajouter un mot dans certains cas, comme si on utilise par exemple un pronom du type possessif (« le mien »). Aussi, il faut se rappeler que le groupe du sujet introduit la locution du type « il faut que » lorsque le groupe du verbe est au subjonctif. Il faut donc additionner les mots appartenant à cette locution (*nMotsSubj*) au nombre de mots du groupe du sujet. Une fois le nombre de mots déterminé, on initialise les vecteurs du groupe du sujet devant en contenir les mots (*motGS* – vecteur de type *String*), les lemmes (*lemmeGS* – vecteur de type *String*), les classes grammaticales (*POSGS* – vecteur de type entier) et les étiquettes morpho-syntaxiques (*POSdGS* – vecteur de type *String*). L'extrait de code 4.24 fournit le détail de ces initialisations.

Une fois les variables de l'extrait de code 4.24 créées, on y insère un à un chacun de leurs éléments, selon la séquence de mots fournie à la Section 3.5.3.2.3. Il faut toutefois y ajouter la locution du type « il faut que » dans le cas du subjonctif. Le détail de cette opération assez triviale mais plutôt longue, n'est pas illustré ici.

4.10.4.2. Création de l'objet du groupe du sujet avec des noms communs

Tel que mentionné précédemment, le groupe du sujet ne se bâtit avec des noms communs que quand le groupe du verbe est à la troisième personne. S'il est au singulier, le groupe du nom ne peut comprendre qu'un seul nom commun. En revanche, si le groupe du verbe est au pluriel, on peut soit utiliser un seul nom commun au pluriel, ou combiner plus d'un nom commun. Pour ce projet, on permet donc l'utilisation de deux noms communs distincts pour former le pluriel. Dans le cas du pluriel, la première étape consiste donc à déterminer au hasard si on utilisera un seul nom au pluriel, ou deux noms. Dans le cas de l'utilisation de deux noms, plusieurs options sont possibles, tel qu'on le montrait au Tableau 3.29. Par exemple, le masculin pluriel peut s'obtenir en combinant un nom commun au féminin pluriel avec un autre nom commun au masculin. On détermine au hasard, basé sur le paramètre arbitraire « *probNoms* », si on fait appel à un nom au pluriel ou à deux noms pour former un groupe du sujet au pluriel avec noms communs.

Ensuite, on détermine au hasard si les noms sont accompagnés d'adjectifs. On détermine aussi au hasard si un adverbe d'intensité (« *très* beau ») modifie l'adjectif. Si deux noms communs sont employés, ces déterminations se font pour chacun des deux mots distinctement. On pourrait par exemple générer « le très beau camion et la voiture », où un seul des deux noms est accompagné d'un adjectif et d'un adverbe. L'extrait de code 4.25 illustre comment ces étapes sont effectuées.

Extrait de code 4.24 : Préparation des variables du groupe du sujet, pour le cas avec pronoms

```
nMotsGS=nMotsSubj+1; // Nombre de mots du subjonctif plus 1 mot pour le pronom
if(casPronom==1|casPronom==3|casPronom==4|casPronom==6) nMotsGS++;
if(casPronom==9)nMotsGS=nMotsGS+2;

// Initialiser les arrays pour objetGS avec bonne taille
motGS   = new String[nMotsGS];
lemmeGS = new String[nMotsGS];
POSGS   = new int   [nMotsGS];
POSdGS  = new String[nMotsGS];
```

Extrait de code 4.25 : Déterminer au hasard le nombre de noms communs, ainsi que la présence d'adjectifs et d'adverbes, pour le groupe du sujet avec noms communs

```
nNoms=1; // par défaut, nombre de noms=1 (en particulier, au singulier)
if(plur){if(Math.random()<prob2noms) nNoms=2;}
// Déterminer au hasard probabilités adjectifs et adverbes pour premier nom
avecAdjectif1=Math.random()<probAdjectif;
if(avecAdjectif1) avecAdverbe1 =Math.random()<probAdverbe;
else avecAdverbe1=false; // Pas d'adverbe si pas d'adjectif!
If(nNoms==2) {
avecAdjectif2=Math.random()<probAdjectif;
if(avecAdjectif2) avecAdverbe2 =Math.random()<probAdverbe;
else avecAdverbe2=false; // Pas d'adverbe si pas d'adjectif!
}
```

Il faut ensuite déterminer au hasard tous les mots à utiliser, soit les noms communs, et les adjectifs et adverbes au besoin. Ces choix sont basés sur les fréquences d'apparition des lemmes dans le corpus de référence, mais aussi sur les cooccurrences des mots déjà incorporés dans la phrase. Ces mots déjà incorporés (noms, adjectifs, verbes et adverbes) sont contenus dans le vecteur *coocPlus*. L'extrait de code 4.26 illustre comment tous ces mots sont déterminés au hasard, dans le cas où un seul nom commun est requis. La procédure est plus longue, mais équivalente, pour le cas où deux noms communs sont requis. Pour sélectionner un adverbe, on utilise la même fonction que celle utilisée pour le groupe du verbe, soit « *ChoisirAdverbe* ». Comme le nombre d'adverbes d'intensité est relativement limité, on ne pose aucune contrainte de cooccurrences. On choisit parmi tous les adverbes d'intensité des banques de données du départ. Pour sélectionner les noms communs et les adjectifs, on fait appel à la méthode « *ChoixNom* ». Malgré son nom, celle-ci est en effet utilisée autant pour sélectionner les noms que les adjectifs, puisque le principe de base est le même. Cette méthode est plus complexe que celle pour choisir un adverbe, puisque les noms choisis doivent satisfaire des contraintes pour le genre et le nombre, ce qui n'était pas le cas pour les adverbes, qui sont invariables. La méthode « *ChoixNom* » est discutée plus en détail à la section suivante. Il est à noter qu'à l'extrait de code 4.26, on s'assure, à chaque nouveau mot incorporé à la phrase, de l'inclure au vecteur *coocPlus*, grâce à la méthode *AjoutCooc* (extrait de code 4.8).

Extrait de code 4.26 : Choix au hasard des noms, adjectifs et adverbes pour le groupe du sujet avec noms communs

```
if(nNoms==1){ // Cas un seul nom
    genre1=genre; nombre1=nombre;
    index1=genre*2+nombre; index2=0; genre2=0; nombre2=0;
    nomFlechi1=ChoixNom(true, genre, nombre, coocPlus);
    coocPlus=Ut.AjoutCooc(coocPlus, nomFlechi1.lemme);
    if(avecAdjectif1){
        adjFlechi1=ChoixNom(false, genre, nombre, coocPlus);
        coocPlus=Ut.AjoutCooc(coocPlus,adjFlechi1.lemme);}
    else adjFlechi1=new MotFlechi();
    if(avecAdverbe1){
        adverbe1=ChoisirAdverbe(coocVide, 41);
        coocPlus=Ut.AjoutCooc(coocPlus,adverbe1);}
} // fin du if un seul nom
```

4.10.4.3. Choix de noms et d'adjectifs au hasard

La même méthode « *ChoixNom* » sert à sélectionner au hasard les noms communs et les adjectifs. Elle accepte justement comme paramètre d'entrée une variable booléenne égale à « TRUE » quand on recherche un nom, ou égale à « FALSE » quand on recherche plutôt un adjectif. On utilise ce paramètre pour mettre en mémoire dans la table de hachage « liste » ou bien la table de hachage de tous les noms communs, ou bien la table de hachage de tous les adjectifs.

La méthode accepte aussi deux entiers, un pour déterminer le genre, l'autre le nombre. Une valeur de « -1 » pour ces paramètres signifie qu'ils peuvent être déterminés au hasard. Des valeurs de « 0 » (masculin) ou « 1 » (féminin) sont autrement spécifiées pour le genre, et le nombre (« 0 » = singulier, « 1 » =pluriel). Finalement, comme pour les méthodes de sélection au hasard des verbes et adverbes, on reçoit en entrée un vecteur *coocPlus* de lemmes déjà inclus dans la phrase, servant à déterminer les cooccurrences parmi lesquelles choisir le mot. Si aucune cooccurrence n'est fournie (la taille du vecteur est de zéro), on choisit alors le mot parmi les tables de hachage complètes. Sinon, on extrait toutes les cooccurrences avec la méthode « *TrouverCooccur* », mais on doit ensuite se limiter aux lemmes correspondant aux noms communs ou adjectifs, le cas échéant, en utilisant une méthode « *Get* » sur l'objet « *classes* » en sortie de « *TrouverCooccur* ».

Mais à ce point, bien que les lemmes de la bonne classe grammaticale soient disponibles, le travail n'est pas terminé, car il faut ensuite se limiter à un sous-ensemble ne comprenant que les versions fléchies correspondant au genre et au nombre recherchés. Il faut se rappeler que les noms communs « non animés » ne se retrouvent qu'au féminin ou qu'au masculin. Par exemple, on dit « une chaise », mais non « un chaise ». Pour les noms animés (« infirmier-infirmière ») et pour les adjectifs, en règle générale, toutes les combinaisons de genres et de nombres sont possibles. Dans le cas où la liste « finale » de noms communs ou adjectifs possibles serait vide, on doit alors utiliser les listes complètes, sans égard aux cooccurrences.

À ce stade, nous obtenons une table de hachage finale, dont les clés de type *String* sont les mots fléchis au bon genre et au bon nombre, et dont les valeurs correspondent aux fréquences de leurs lemmes respectifs, soit au sein des cooccurrences, ou au sein du corpus de référence, le cas échéant. On utilise alors la méthode « *motHasard* » utilisée précédemment (extrait de code 4.10), qui accepte en entrée notre table de hachage, ainsi que les paramètres fournissant la méthode pour appliquer le hasard. La méthode « *ChoixNom* » est illustrée dans son entièreté à l'extrait de code 4.27. Elle fournit en sortie l'objet *motFlechiSortie*, qui contient les variables pour le mot fléchi et son lemme correspondant.

Extrait de code 4.27 : Méthode « *ChoixNom* » pour déterminer au hasard les noms communs et adjectifs, en fonction de leur genre et nombre

```
public MotFlechi ChoixNom(boolean nom, int genre, int nombre, String [] cooc){

boolean affiche=false; boolean ambigu=false; OccurObjet classes;
MotFlechi motFlechiSortie = new MotFlechi();
String lemme, motFlechi; int tailleGN;
HashMap<String,String[]> motLocal;  //=new HashMap<String,String[]>();
HashMap<String,Integer>  liste;     //   =new HashMap<String,Integer>();
HashMap<String,Integer>  listeFin;  // =new HashMap<String,Integer>();
HashMap<String,Integer>  listeGN =new HashMap<>();
HashMap<String,Integer>  listeGN2;
if(genre==-1) {if(Math.random()<0.5)genre =0; else genre =1;}
if(nombre==-1){if(Math.random()<0.5)nombre=0; else nombre=1;}
int index=genre*2+nombre;
if(nom){ // On choisit un nom
    motLocal=nomsGlobal;
    if(ambigu) liste = nomsA; else liste = nomsNA;
}
else{ // On choisit un adjectif
    motLocal=adjectifsGlobal;
    if(ambigu) liste = adjectifsA; else liste = adjectifsNA;
}
int tailleCooc=cooc.length;
if (tailleCooc<1) listeFin=liste;
else{ // Cooccurrences fournies
    HashMap<String,Integer>[] arrayTables;
    arrayTables = new HashMap[cooc.length];
    for(int i=0;i<cooc.length;i++){
        classes=TrouverCooccur(cooc[i]);
        if(nom) arrayTables[i]=classes.GetNoms();
        else    arrayTables[i]=classes.GetAdjectifs();
    } // fin du for i
    listeFin=Ut.CombineFreq(arrayTables);
    if(listeFin.isEmpty()) listeFin=liste;
} // fin du else cooccurrences fournies
Set<String> keys = listeFin.keySet();
for(int i=0;i<keys.size();i++){ // passer à travers tous les lemmes de la table
    String lemmeBoucle=(String)keys.toArray()[i];  // Extraire le lemme
    int freq=listeFin.get(lemmeBoucle);            // Extraire sa fréquence
    String [] forme;
    forme=motLocal.get(lemmeBoucle);
    if(!forme[index].equals("!")) listeGN.put(lemmeBoucle, freq);
} // fin du for i keys size
tailleGN=listeGN.size();
if(tailleGN==0){
    if(nom){
        if(index==0)listeGN=nomGN0; if(index==1)listeGN=nomGN1;
        if(index==2)listeGN=nomGN2; if(index==3)listeGN=nomGN3;
    }
    else{
        if(index==0)listeGN=adjGN0; if(index==1)listeGN=adjGN1;
        if(index==2)listeGN=adjGN2; if(index==3)listeGN=adjGN3;
    }
}
listeGN2=Ut.CoocMemePhrase(listeGN, cooc);
if(!listeGN2.isEmpty()) listeGN=listeGN2;
lemme=Ut.MotHasard(listeGN,1,0.0); motFlechi=motLocal.get(lemme)[index];
return motFlechiSortie;
}
```

Extrait de code 4.28 : Déterminer au hasard l'un des 14 types de déterminants du Tableau 3.30 à utiliser, après avoir assigné un poids (probabilité relative) à chacun

```
// Cas n'impliquant aucune contrainte
nomPoids.put(0,10.0); nomPoids.put(1, 10.0); nomPoids.put(2,10.0);
nomPoids.put(3,80.0); nomPoids.put(10,10.0);

// Cas impliquant des contraintes
if((sing&pos)|(nNoms==2&nombre1==0&nombre2==0)){
   nomPoids.put(4,1.0);  nomPoids.put(9,10.0);
   nomPoids.put(11,10.0); nomPoids.put(12,10.0);}
if((plur&nNoms==1)|(nNoms==2&nombre1==1&nombre2==1)){
   nomPoids.put(5,1.0);  nomPoids.put(6,1.0);
   nomPoids.put(8,1.0);  nomPoids.put(13,1.0);}
if((sing&neg)|(nNoms==2&nombre1==0&nombre2==0)) nomPoids.put(7,1.0);

// Déterminer "cas avec nom" au hasard parmi les 14
casNom=Ut.DoubleHasard(nomPoids, 1, 1.0);
```

4.10.4.4. Choix des déterminants au hasard

Plusieurs options de déterminants sont proposées pour ce projet. Celles-ci ont été décrites au Tableau 3.30, accompagnées de certaines de leurs contraintes, en ce qui a trait au nombre (singulier ou pluriel) et à la forme affirmative ou négative du groupe du verbe. Pour déterminer lequel des 14 types de déterminants est choisi, on assigne arbitrairement un poids à chacun. Des énoncés conditionnels (« IF ») sont ensuite exécutés pour éliminer les cas soumis à des contraintes. Par exemple, on n'utilise le déterminant « chaque » qu'au singulier et le déterminant « nul » qu'à la forme négative. On élimine les cas non permis en remplaçant leur poids par défaut par une valeur de zéro. De la même façon qu'on l'avait fait pour déterminer au hasard les pronoms pour le groupe du sujet avec pronoms, on utilise la fonction « *DoubleHasard* », à laquelle on envoie une table de hachage dont les clés sont les codes des 14 types de déterminants (valeurs entières) et les valeurs sont les poids fournis arbitrairement (valeurs doubles). L'extrait de code 4.28 donne le détail des opérations d'assignations de poids et de la détermination au hasard du type de déterminant choisi. Une fois le type de déterminant choisi avec l'extrait de code 4.28, il faut ensuite s'assurer d'utiliser la version fléchie du déterminant correspondant au bon genre et au bon nombre.

4.10.4.5. Création de compléments du nom

Comme on le mentionnait aux Sections 3.5.3.2.4 à 3.5.3.2.6, lorsqu'il est bâti sur la base d'un ou de noms communs, le groupe du sujet peut aussi incorporer des compléments du nom. Trois types de compléments du nom avaient été présentés : le type « du », le type « qui » et le type « que ». Des algorithmes plus détaillés pour la formation de ces compléments du nom sont présentés ici.

Pour bâtir le complément du nom de type « du », on s'assure en premier lieu que le groupe du sujet est formé sur la base d'un ou de deux noms communs, et non sur la base d'un pronom. Le cas échéant, on détermine ensuite aléatoirement si un complément du nom de type « du » sera introduit, selon un poids arbitraire. Si tel est le cas, on détermine au hasard un nom commun ou deux noms communs pour le complément du nom. De cette information, en découle le genre et le nombre. En fonction du genre et du nombre du nom ou des noms sélectionnés pour le

complément du nom, on impose ensuite l'une des trois formes « du », « de la » ou « des ». La Figure 4.44 résume le processus menant à la génération du complément du nom de type « du ».

L'extrait de code 4.29 porte sur la formation du complément du nom de type « du », plus précisément sur la dernière étape illustrée à la Figure 4.44. On y constate que si un tel complément doit être créé (si la variable booléenne « *cNom* » est égale à TRUE), on lance un appel à la méthode « *GroupeNom* », qui contient certains paramètres, incluant la chaîne de caractères « de », informant du besoin de joindre cette préposition au groupe du nom formé. La préposition « de » deviendra en fait soit « du », « de la » ou « des », selon le genre et le nombre, tel qu'illustré à la Figure 4.44. La méthode « *CompNomDu* » se retrouve donc à être très courte, puisqu'elle se base sur une autre méthode existante servant à former un groupe du nom.

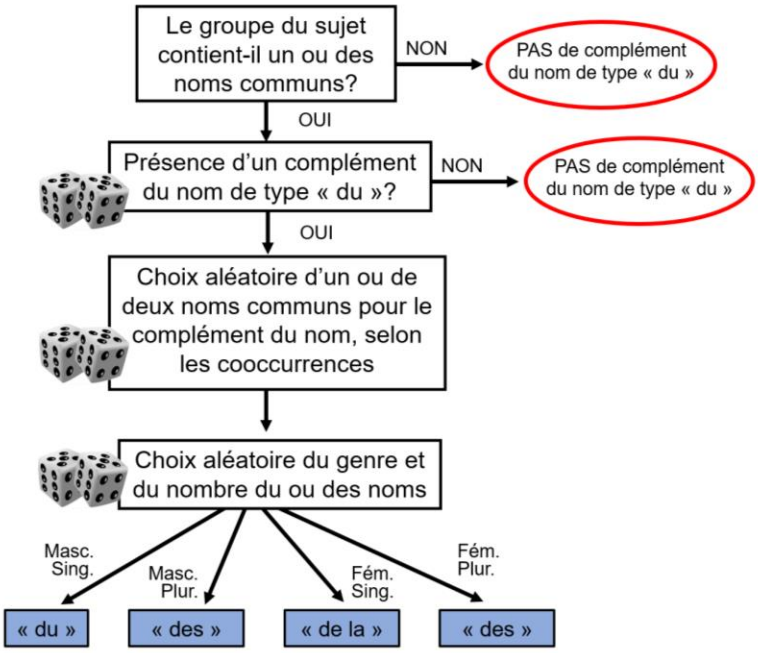


Figure 4.44 : Processus pour générer un complément du nom de type « du »

Extrait de code 4.29 : Méthode pour générer le complément du nom de type « du »

```
public ObjetGS CompNomDu(boolean cNom, ObjetGV objetGV, String[] cooc){

ObjetGS objetCNDu;
if(cNom) objetCNDu = GroupeNom(-1, -1, 3, -1, false, 3, "de", objetGV, cooc);
else{ // Créer un objet vide
    objetCNDu           = new ObjetGS(0, 0);
    objetCNDu.motsGS    = new String[0];
    objetCNDu.lemmesGS  = new String[0];
    objetCNDu.POSGS     = new int   [0];
    objetCNDu.POSdGS    = new String[0];
    objetCNDu.coocGS    = cooc;
}
return objetCNDu;
}
```

Le processus menant quant à lui à la détermination des compléments du nom de type « qui » et de type « que » est illustré à la Figure 4.45. On y constate qu'on détermine d'abord aléatoirement, selon un poids arbitraire, si le groupe sujet inclura ou non un complément du nom de type « qui » ou de type « que ». Le cas échéant, on sélectionne ensuite un infinitif au hasard, en fonction de la fréquence des verbes parmi les cooccurrences. Ensuite, on vérifie de quel type de verbe il s'agit, selon la classification fournie dans le Bescherelle, à savoir « transitif direct » (« T »), transitif indirect (« Ti »), transitif indirect suivi d'un infinitif (« Ti+inf ») ou intransitif (« I »). Dans les rares cas de verbes ne répondant à aucun de ces critères, on choisit aléatoirement un autre verbe. En fonction du type de verbe, on peut alors déterminer le cas à considérer pour bâtir le complément du nom (« qui », « que », « dont », etc.). La Figure 4.45 inclut des exemples pour chacun des cas. Davantage de détails concernant les algorithmes en place pour les types de verbes (« T », « Ti », « Ti+inf » et « I ») sont présentés à la Section 4.10.5 portant sur le groupe complément.

La méthode Java servant à bâtir les compléments du nom de type « que » ou « qui » est bien plus longue et plus complexe que la très courte méthode pour le complément du nom de type « du » affichée à l'extrait de code 4.29. Elle ne sera donc pas fournie dans son intégralité. Seules quelques étapes charnières seront explicitement démontrées. À la Figure 4.45, on constate tout d'abord que dans les deux cas, il faut en premier lieu choisir un verbe (infinitif) au hasard. L'extrait de code 4.30 fournit les énoncés Java nécessaires à ce choix aléatoire, afin de répondre aux contraintes exprimées à la Figure 4.45.

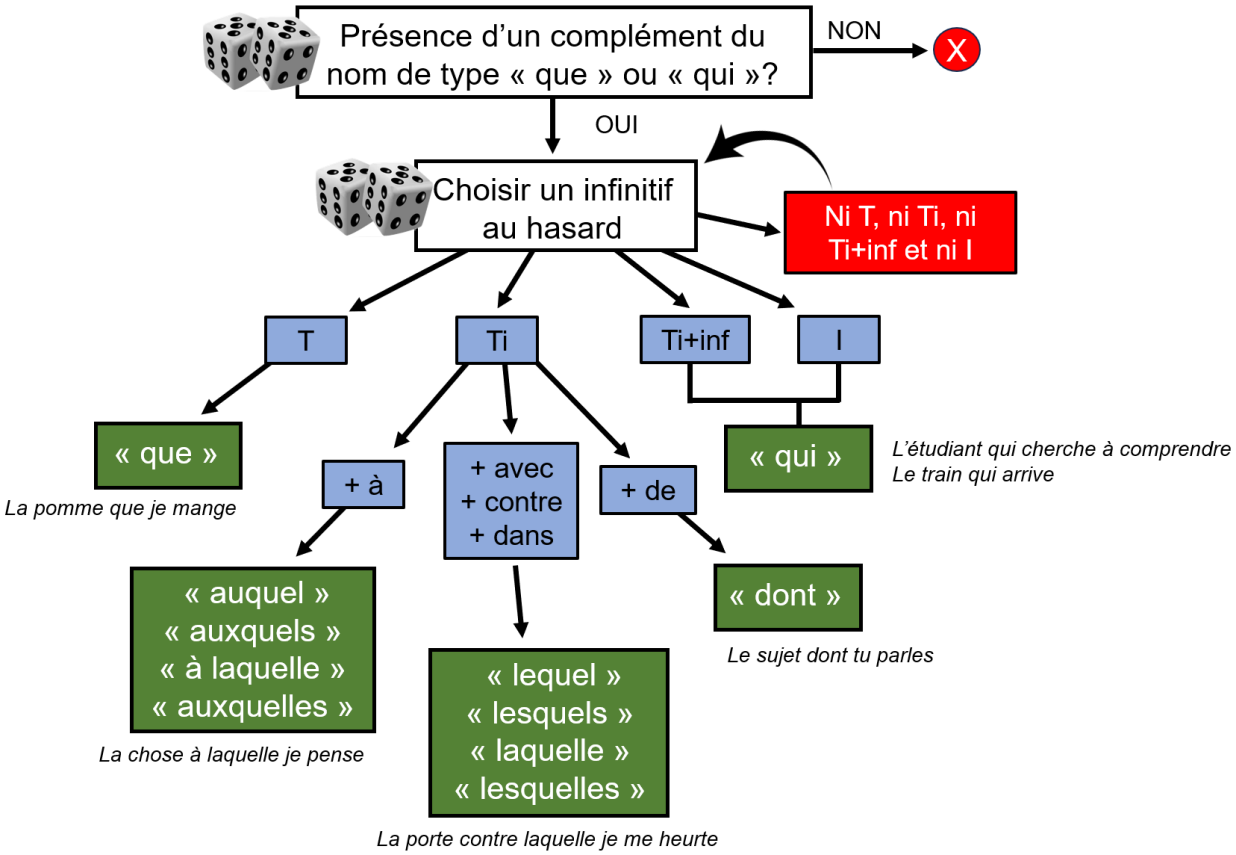


Figure 4.45 : Détermination des compléments du nom des types « qui » et « que »

Extrait de code 4.30 : Choix aléatoire du verbe pour le complément du nom de type « que » ou « qui »

```
String type="";
while(type.equals("")){ // Trouver un verbe qui est au moins un T, Ti ou I
    verbeInf = VerbeHasard(false, CooC.global); // Choix de l'infinitif au hasard
    type = VerbeCN(verbeInf);
}
CooC.global=Ut.AjoutCooc(CooC.global, verbeInf);
```

On détermine ensuite, tel que décrit à la Figure 4.45, si on ira de l'avant avec un complément de type « qui » ou un complément de type « que », selon le type de verbe, et finalement selon le hasard, dans le cas où les deux options sont permises. L'extrait de code 4.31 décrit les énoncés Java utilisés à cette fin.

Si le type de verbe ou le hasard fait en sorte qu'on opte pour un complément du nom de type « qui » (variable booléenne « qui » est égale à TRUE), les énoncés listés à l'extrait de code 4.32 sont effectués. On constate dans cet extrait qu'on doit tenir compte du nombre associé au groupe nom. En effet, dans le cas singulier, le verbe sera conjugué à la troisième personne du singulier. Dans le cas pluriel, le verbe sera conjugué à la troisième personne du pluriel. Le genre du groupe du nom peut aussi s'avérer nécessaire, dans le cas des temps composés faisant appel au participe passé. Un paramètre « *modeCode* » est aussi créé selon le type du verbe (transitif, intransitif, etc.). Ce paramètre est ensuite passé à l'appel pour la méthode « *GroupeVerbe* », car le groupe du verbe dépend en effet du type de verbe.

Le complément du nom de type « qui » nécessite aussi un complément. On lance donc un appel à la méthode « *GroupeComplement* ». On ne s'attarde pas à cette méthode pour l'instant, puisqu'elle sera décrite plus en détail à la Section 4.10.5. On se contentera ici de préciser que les paramètres passés à cette méthode sont en lien avec le verbe employé *dans le complément du nom*, et non avec le verbe principal employé au groupe du verbe.

Finalement, une fois le « groupe verbe » et le « groupe complément » du complément du nom de type « qui » déterminés, il ne reste qu'à faire débuter ce segment de phrase par le mot « qui » lui-même, comme on le voit aux derniers énoncés de l'extrait de code 4.32.

Extrait de code 4.31 : Déterminer si le complément sera de type « que » ou de type « qui ». En sortie, la variable booléenne « qui » fournit ce choix

```
qui = false;

// Imposer le qui dans certains cas
if(type.equals("I")|type.equals("à+inf")|type.equals("de+inf")) qui=true;

// Verbe se conjugue-t-il avec auxiliaire être? (exclusivement ou non)
boolean etre = infinitifH.get(verbeInf).GetDonnees()[6];

// Imposer le qui aussi, si verbe se conjugue avec auxiliaire être
if(etre) qui=true;

// Déterminer au hasard si option "qui" dans les cas autres que ceux ci-haut
if(!qui) qui = Math.random()<Parametres.probQui;
```

Extrait de code 4.32 : Énoncés Java pour bâtir le complément du nom de type « qui »

```
// Commencer par définir le groupe verbe (pas impératif)
Parametres.casQuiDansVerbe=true;
int personne; if(nombre==1)personne=6; else personne=3;
int modeCode=0;
if(type.equals("I"))  modeCode=2;
if(type.equals("T"))  modeCode=1;
if(!type.equals("I")&!type.equals("T")&!type.equals("")){
    modeCode=5; Parametres.prepCN=type;
}

Parametres.genre=genre; // Si conjugué avec aux être, genre sera correct
objetGVCN = GroupeVerbe(verbeInf, -1, personne, -1, modeCode, CooC.global);
CooC.global=objetGVCN.coocGV;
Parametres.casQuiDansVerbe=false;
Parametres.genre=-1; // Retour à la valeur par défaut
Parametres.deDesPossible=(objetGVCN.negatifGV&&modeCode==1);

// Définir le groupe complément de ce groupe verbe du CN
if(modeCode!=2)objetGVCN.intransitifGV=false;
objetGCCN = GroupeComplement(genre, nombre, objetGVCN.attributifGV,
          objetGVCN.intransitifGV, objetGVCN.prepositionGV, objetGVCN.tempsGV,
          objetGVCN.negatifGV, objetGVCN.personneGV, objetGVCN, CooC.global);
CooC.global=objetGCCN.coocGC;

// Ajouter le mot "qui" - Définir l'objet pour "qui"
objetQui   = new ObjetMotLemmePOS();
String [] mot = new String[1]; mot[0]="qui";
objetQui.mot=mot; objetQui.lemme=mot;
int [] POS = new int[1]; POS[0]=6;
String [] POSd = new String[1]; POSd[0] = CodePOS.Code(6, 0, 0, 0, 0);
objetQui.POS=POS; objetQui.POSd=POSd;
Parametres.deDesPossible=false;
```

Comme on le mentionnait au Chapitre 3 et comme on le voit à la Figure 4.45, les compléments du nom de type « que » regroupent en fait plusieurs possibilités de préposition (« que », « dont », « avec lequel », etc.). Il faut donc sélectionner la ou les prépositions requises, en fonction du verbe choisi. Ce choix se fait en fonction du type de verbe, transitif direct ou indirect. Dans le cas de transitifs indirects, le choix de préposition dépend des possibilités fournies dans le tableau du Bescherelle pour le verbe en question. L'extrait de code 4.33 fournit les énoncés Java servant à déterminer les prépositions à utiliser et les variables et vecteurs se rapportant à ce choix.

Finalement, l'extrait de code 4.34 fournit les énoncés visant à bâtir toutes les variables et vecteurs requis pour le complément du nom de type « que ». On doit y spécifier la personne du verbe (qui dépend du nombre du groupe du nom employé), ainsi que certains paramètres qui imposent certaines contraintes quant aux choix de temps de verbe. Le complément du nom de type « que », comme on le voit à l'extrait de code 4.34, regroupe donc un groupe du verbe et un groupe du nom, auxquels on ajoute la ou les prépositions.

Extrait de code 4.33 : Déterminer les prépositions dans le cas de complément de type « que »

```
nPrep=1; // Nombre de prépositions requises (valeur par défaut)

// Vérifier si le cas requiert 2 prépositions (e.g. "avec lequel")
if((type.equals("à")&index==2)|(type.equals("avec")|type.equals("contre")|
    type.equals("dans")|type.equals("sur")|type.equals("vers")|
    type.equals("pour"))){
    nPrep=2;
}
// Définir les vecteurs selon le nombre de prépositions
String [] motPrep   = new String[nPrep];
String [] lemmePrep = new String[nPrep];
int    [] POSprep   = new int   [nPrep];
String [] POSdPrep  = new String[nPrep];

// Cas transitif, on emploie "que"
if(type.equals("T")){
    motPrep[0]="que"; lemmePrep[0]="que"; POSprep[0]=6;
    POSdPrep[0] = CodePOS.Code(6, 0, 0, 0, 0);
    Parametres.reglePP=true; // pp avec avoir s'accorde avec COD devant verbe
    Parametres.indexPP=index; // index du pp quand reglePP = true
}
if(type.equals("à")){
    POSprep[0]=6; POSdPrep[0] = CodePOS.Code(6, 0, 0, 0, 0);
    switch(index) {
        case 0 ->   {motPrep[0]="auquel";   lemmePrep[0]="auquel";}
        case 1 ->   {motPrep[0]="auxquels"; lemmePrep[0]="auquel";}
        case 2 ->   {
            motPrep[0]="à"; lemmePrep[0]="à"; POSprep[0]=7;
            POSdPrep[0] = CodePOS.Code(7, 0, 0, 0, 0);
            motPrep[1]="laquelle"; lemmePrep[1]="lequel"; POSprep[1]=6;
            POSdPrep[1] = CodePOS.Code(6, 0, 0, 0, 0);
       }
       default ->  {motPrep[0]="auxquelles"; lemmePrep[0]="auquel";} // (Case 3)
    } // Fin du switch
}
if(type.equals("de")){
    motPrep[0]="dont"; lemmePrep[0]="dont"; POSprep[0]=6;
    POSdPrep[0] = CodePOS.Code(6, 0, 0, 0, 0);
}
if(type.equals("avec")|type.equals("contre")|type.equals("dans")|
   type.equals("sur")|type.equals("vers")){
    motPrep[0]=type; lemmePrep[0]=type;
    POSprep[0]=7; POSdPrep[0] = CodePOS.Code(7, 0, 0, 0, 0);
    POSprep[1]=6; POSdPrep[1] = CodePOS.Code(6, 0, 0, 0, 0);
    motPrep[1] = switch (index) {
        case 0  -> "lequel";
        case 1  -> "lesquels";
        case 2  -> "laquelle";
        default -> "lesquelles";
    }; // Fin du switch
    lemmePrep[1]="lequel"; // Version de base
} // Fin du if type equals variés
```

Extrait de code 4.34 : Bâtir les éléments du complément du nom de type « que »

```
// Singulier ou pluriel au hasard pour le verbe (sujet déterminé selon verbe)
boolean pluriel = Math.random()>0.5;

int personne; // Seules options: 3 et 6, par choix/contrainte
int nombreN; // Variable pour le nombre du nom qui accompagne le verbe dans ce CN
if(pluriel) {personne=6; nombreN=1;} else {personne=3; nombreN=0;}

Parametres.casQue=true; // Exclure certains temps de verbe pour le CN
Parametres.casCN=true;  // Empêche de choisir l'impératif

// Appel au groupe verbe. Temps et négatif au hasard (=-1). Mode au hasard (=0)
objetGVCN=GroupeVerbe(verbeInf, -1, personne, -1, 0, CooC.global);
CooC.global = objetGVCN.coocGV;

// Appel au groupe du nom - nombre selon groupe verbe du CN, genre au hasard
objetGNCN=GroupeNom(-1, nombreN, 1, -1, false, personne, "", objetGVCN,
CooC.global);
CooC.global = objetGNCN.coocGS;

// Ajouter les "prépositions" au groupe du nom, à l'indice 0
GNCN=Ut.AjouteMot(objetGNCN.motsGS, objetGNCN.lemmesGS, objetGNCN.POSGS,
objetGNCN.POSdGS, motPrep, lemmePrep, POSprep, POSdPrep, 0, "");
```

4.10.4.6. Préparation de l'objet ObjetGS pour le cas avec noms communs

Pour bâtir le groupe du sujet avec noms communs, on doit déterminer le nombre total de mots (*nMotsGS*) qu'il doit inclure. Ce nombre dépend du nombre de noms communs utilisés (un ou deux), du type de déterminant car certains impliquent plus d'un mot (« *un tel* camion », « *n'importe quelle* voiture »), de la présence d'adjectifs et d'adverbes, ainsi que de la présence potentielle de compléments du nom. Aussi, il faut se rappeler que le groupe du sujet introduit la locution du type « il faut que » lorsque le groupe du verbe est au subjonctif. Il faut donc additionner les mots appartenant à cette locution (*nMotsSubj*) au nombre de mots du groupe du sujet. Une fois le nombre de mots déterminé, on initialise les vecteurs du groupe du sujet devant en contenir les mots (*motGS* – vecteur de type *String*), les lemmes (*lemmeGS* – vecteur de type *String*), les classes grammaticales (*POSGS* – vecteur de type entier) et les étiquettes morpho-syntaxiques (*POSdGS* – vecteur de type String).

4.10.4.7. Détermination de l'objet ObjetGS

Finalement, une fois tous les vecteurs du groupe du sujet initialisés pour le cas avec pronoms ou avec noms communs, on doit préparer l'information complète qui doit être fournie en sortie du groupe du sujet. Cette information servira ensuite à bien arrimer le groupe du complément dans le cas de verbes attributifs, et d'ultimement bâtir la phrase complète. Comme une méthode ne peut fournir qu'un seul élément en sortie (commande *return*), on crée un objet (classe *ObjetGS*) contenant toute cette information, et c'est cet objet qui est retourné par la méthode *GroupeSujet*. Cette classe est illustrée à l'extrait de code 4.35.

Cette classe comprend un constructeur qui initialise les vecteurs selon la dimension requise. Elle comprend ensuite une méthode de type « *Set* » qui sert à assigner toutes les valeurs à toutes les variables de la classe (*SetGS*). Certaines variables de la méthode *SetGS* servent de paramètres d'entrée pour la méthode servant à bâtir le groupe du complément dans le cas d'un verbe attributif,

où le complément doit s'accorder avec le groupe du sujet. Cette classe ne comprend pas de méthode de type « *Get* ». Les variables de cette classe publique seront donc extraites directement par des références du type *ObjetGS.nomdevariable* lorsque requises au sein des autres méthodes.

Extrait de code 4.35 : Classe « *ObjetGS »*, fournissant l'objet à fournir en sortie de la méthode du groupe du sujet. Elle comprend un constructeur et une méthode de type « *Set* »

```
public class ObjetGS {

int nMotsGS, nCoocGS, genreGS, nombreGS;
String texteGS;
String[] motsGS, lemmesGS, coocGS, POSdGS;
int [] POSGS;
boolean enleverPas, avecNomGS;

public ObjetGS(int nMots, int nCooc){ // CONSTRUCTEUR

nMotsGS=nMots;
nCoocGS=nCooc;
motsGS   = new String[nMotsGS];
lemmesGS = new String[nMotsGS];
POSdGS   = new String[nMotsGS];
coocGS   = new String[nCooc]  ;
POSGS    = new int   [nMotsGS];
}

public void SetGS(String texte,  String [] mots,  String [] lemmes, String []
                  POSd, int [] POS, int genre, int nombre,
                  boolean enleverPasIn, String [] cooc, boolean avecNom){

texteGS=texte;
motsGS = mots; lemmesGS=lemmes; coocGS=cooc;
POSGS=POS;
POSdGS=POSd;
nombreGS=nombre;
genreGS=genre;
enleverPas=enleverPasIn;
avecNomGS=avecNom;
}
}
```

4.10.5. Création de l'objet du groupe du complément

La forme à adopter pour le groupe du complément dépend du « mode » associé au verbe sélectionné dans le groupe du verbe, tel que discuté à la Section 3.5.3.3. Les mêmes modes discutés au Chapitre 3 sont de nouveau discutés ici, mais dans le contexte de la programmation de l'algorithme. Cependant, la séquence dans laquelle on les aborde n'est pas la même, pour des raisons pratiques en lien avec le fonctionnement de l'algorithme. En effet, on débute avec le cas « transitif indirect avec infinitif », puisque celui-ci peut par la suite être suivi d'un autre élément de complément parmi les autres.

4.10.5.1. Cas transitif indirect avec infinitif

On débute la méthode du groupe du complément en vérifiant en premier lieu si le groupe du verbe demande une préposition suivie d'un infinitif. Cette vérification s'effectue tout simplement en vérifiant la préposition fournie en sortie dans l'objet *ObjetGC.* Dans un cas où aucune préposition n'est requise, la variable contenant la préposition est vide. Dans le cas qui nous intéresse ici, on vérifie que la préposition contient les caractères « *+inf* », donc qu'elle demande à être suivie par un infinitif (« je tente de *trouver* »), plutôt que par un groupe nominal (« je demande à *un ami* »).

Le complément dans un tel cas débute avec la préposition fournie par le groupe du verbe. On enchaine ensuite avec un verbe à l'infinitif. Mais par convenance et par simplicité, on exclut les verbes demandant une forme pronominale et ceux demandant un transitif indirect avec infinitif, tel qu'expliqué au Chapitre 3. Pour choisir l'infinitif, on emploie donc une boucle « *while* » qui choisit un infinitif au hasard parmi tous les verbes non-ambigus et qui itère jusqu'à ce qu'un infinitif répondant à ces contraintes soit identifié (extrait de code 4.36).

Extrait de code 4.36 : Boucle « *while* » servant à sélectionner au hasard un infinitif qui ne demande pas de forme pronominale ou un transitif indirect avec infinitif

```
while(!verbeOK){
infinitif = VerbeHasard(false, coocVide); // choix parmi verbes non ambigus
ObjetVerbe     = infinitifH.get(infinitif);
verbeBoolean   = ObjetVerbe.GetDonnees();
verbePrep      = ObjetVerbe.GetPrep();
attributif     = verbeBoolean[9];
modal          = verbeBoolean[10];
T_direct       = verbeBoolean[0];
T_indirect     = verbeBoolean[4];
intransitif    = verbeBoolean[1];
prepOK=false;
for(int i=0;i<3;i++){
   if(!verbePrep[i].contains("+")&!verbePrep[i].equals("")){
      prepOK=true;preposition=verbePrep[i];}}
      // Cette boucle choisit la dernière préposition sans +inf
      if(intransitif)       verbeOK=true;
      if(T_direct)         {verbeOK=true; preposition="";}
      if(T_indirect&prepOK) verbeOK=true;
      if(modal)             verbeOK=false;
}
}
```

Extrait de code 4.37 : Détermination de l'adjectif et de l'adverbe (le cas échéant) dans le cas d'un verbe attributif

```
motAdjectif=ChoixNom(false, genre, nombre, coocPlus);
avecAdverbe = Math.random()<probAdverbe;
if(avecAdverbe) adverbe=ChoisirAdverbe(arrayVide, 41);
```

L'avantage de débuter avec le transitif indirect avec infinitif, est qu'une fois que la préposition et l'infinitif sont définis, on peut poursuivre avec le reste de la méthode et ajouter d'autres mots au groupe du complément, en fonction du verbe à l'infinitif choisi. Par exemple, on pourrait avoir :

- Il tente de *marcher*. (le verbe « marcher » est intransitif – on arrête là)
- Il tente de *gagner la partie*. (le verbe « gagner » accepte un transitif direct)
- Il tente de *partir à la campagne*. (le verbe « partir » accepte un transitif indirect sans infinitif)
- Il tente de *devenir fort*. (le verbe « devenir » est un attributif, demande un adjectif)

On poursuit donc avec les tests pour les autres modes, non plus sur la base du mode initial (transitif indirect avec infinitif), mais plutôt selon le mode du verbe infinitif choisi à l'extrait de code 4.36. On assigne toutefois la valeur « TRUE » à une variable booléenne « *TiInf* » servant à se rappeler de bien inclure la préposition originale et le verbe à l'infinitif au groupe du complément, une fois qu'on s'attarde aux autres options pour poursuivre le complément.

4.10.5.2. Cas attributif

Dans le cas d'un verbe attributif, le groupe du complément contient un adjectif ou un participe passé employé seul. Mais l'adjectif ou participe passé doit s'accorder en genre et en nombre avec le groupe du sujet. Les variables « genre » et « nombre » en sortie du groupe du sujet ont donc été envoyées en entrée du groupe du complément pour permettre cet accord. De plus, on permet la possibilité d'inclure un adverbe d'intensité. On a donc deux options possibles :

- La cliente devient *fâchée*. (sans adverbe)
- La cliente devient *très fâchée*. (avec adverbe)

L'utilisation d'un adverbe est déterminée au hasard, en fonction du paramètre « *probAdverbe* » dont la valeur entre « 0 » et « 1 » est arbitraire. L'adjectif et l'adverbe (le cas échéant) sont déterminés au hasard avec les commandes listées à l'extrait de code 4.37.

L'appel de la méthode « *ChoixNom* » inclut la valeur booléenne « FALSE » pour indiquer qu'on recherche ici un adjectif, et non un nom commun. On impose ensuite le genre et le nombre en fonction du sujet, tel que discuté plus haut. On passe aussi le vecteur des lemmes utilisés jusqu'à présent dans la phrase (*coocPlus*), afin de choisir parmi les cooccurrences de ces lemmes. Pour l'adverbe, on impose ici un adverbe d'intensité pour accompagner l'adjectif, associé au code « 41 », mais on n'impose pas de cooccurrence, puisque les adverbes d'intensité ne sont pas si nombreux. Les algorithmes et codes pour déterminer les adjectifs et les adverbes ont été présentés précédemment.

En général, le groupe du complément dans le cas d'un verbe attributif ne comprend donc qu'un seul mot ou deux mots, selon la présence d'un adverbe, comme dans les deux exemples montrés plus haut. Cependant, il ne faut pas oublier qu'il se peut que le groupe du complément débute avec une préposition suivie d'un infinitif demandant lui-même un attributif, quand la variable « *TiInf* » a la valeur TRUE. On pourrait par exemple avoir :

- La cliente tend *à devenir fâchée*.
- La cliente tend *à devenir très fâchée*.

Dans un tel cas, le groupe du complément comporte alors ou bien trois mots (sans adverbe) ou bien quatre mots (avec adverbe).

4.10.5.3. Cas intransitif

Le cas avec verbe intransitif est le plus simple, puisqu'un tel verbe ne requiert aucun complément. On a par exemple :

- Le chien *aboie*. (aucun complément du verbe)

Dans un tel cas, l'objet en sortie du groupe du complément reste vide. Cependant, il ne faut pas oublier qu'il se peut que le groupe du complément débute avec une préposition suivie d'un infinitif qui lui serait intransitif. On pourrait par exemple avoir :

- Le chien tend *à aboyer*.

Dans cet exemple, le verbe « tendre » est transitif indirect avec infinitif, mais l'infinitif « aboyer » quant à lui est intransitif, et n'accepte pas de complément. Dans un tel cas, le groupe du complément comprend donc deux mots, soit la préposition et l'infinitif intransitif.

4.10.5.4. Cas transitif direct

On utilise le transitif direct quand le verbe demande un complément d'objet direct, ne faisant donc pas appel à une préposition. Dans un tel cas, le groupe du complément devient très semblable au groupe du sujet discuté à la Section 4.10.4. Par contre, on ne peut utiliser exactement la même méthode, puisque :

- On n'a pas à se soucier du verbe subjonctif pour le complément, comme on devait le faire pour le groupe du sujet (voir Section 3.5.3.2)
- Par simplicité et convenance, on ne permet pas l'utilisation de pronoms dans le cas où ce transitif direct suit un transitif indirect avec infinitif. Par exemple, bien que cette phrase soit tout à fait correcte, on ne la permet pas pour ce projet :
    - Le chien tend à la manger.
- Quand on emploie un pronom tel que « le », « la », ou « les », ce pronom doit être déplacé devant le verbe :
    - Le chien ronge un os. → Il *le* mange.
    - Le chien regarde la chatte. → Il *la* regarde.
    - Le chien abime les rideaux. → Il *les* abime.

Une méthode distincte de celle du groupe du sujet a donc dû être créée pour le cas transitif direct, qu'on a appelée ici « *GroupeNom* ». En entrée de cette méthode, on doit donc spécifier si le complément fait suite à un verbe demandant une préposition suivie d'un infinitif, pour tenir compte des contraintes décrites ci-haut.

Tout comme pour le groupe du sujet, on permet d'employer soit des noms, soit des pronoms. Quand on emploie des noms, on n'en utilise qu'un pour le singulier, et on en utilise un ou deux pour le pluriel (voir le Tableau 3.29). Le choix du pluriel ou du singulier se fait au hasard, selon un seuil arbitraire. Il n'est pas nécessaire ici de s'arrimer avec le groupe du sujet, comme pour le cas d'un verbe attributif.

On peut donc se retrouver avec des phrases comme celles-ci :

- Le chien fait une marche. (au singulier)
- Le chien fait une très grande marche. (avec adjectif et adverbe)
- Le chien regarde les chats. (au pluriel)
- Le chien le mange. (avec pronom, déplacé devant le verbe)
- Le chien mange celui-ci. (avec pronom, après le verbe)
- Le chien n'en mange aucun. (avec pronom, pour cas négatif)

Comme la méthode « *GroupeNom* » est très semblable à la méthode « *GroupeSujet* », le détail de son fonctionnement n'est pas mentionné ici. En revanche, il est important de mentionner la distinction importante concernant le complément, en ce qui concerne le positionnement des pronoms personnels, tel que mentionné plus haut avec l'exemple du chien qui ronge un os. Ceci est le sujet de la prochaine sous-section (Section 4.10.5.4.1).

4.10.5.4.1. Modification du groupe du verbe pour pronoms compléments d'objet directs

Comme on l'a vu à la section précédente, quand le complément d'objet direct est un pronom-objet (me, te, le, nous, vous, les), ce qui correspond à la forme accusative, on doit déplacer ce pronom *devant* le verbe. Dans le cadre du présent projet, cela correspond à laisser le groupe du complément vide, et de plutôt modifier le groupe du verbe pour y insérer le pronom en question. Certains exemples ont été fournis à la section précédente. Mais il faut aussi tenir compte de plusieurs possibilités de positionnement du pronom-objet, tel que discuté à la Section 3.5.3.3. Un exemple de chaque type est répété ici :

Non-modal, Affirmatif :	Je *le* mange.
Non-modal Négatif :	Je ne *le* mange pas.
Modal :	Je vais *le* manger. Je ne vais pas *le* manger.
Impératif positif :	Mange-*le*!
Impératif négatif :	Ne *le* mange pas!

Alors qu'on génère le groupe du complément, il faut donc « retourner en arrière » et modifier le groupe du verbe qu'on avait pourtant complété plus tôt. Cela exige que l'objet *ObjetGV* retourné par la méthode du groupe du verbe soit fourni en entrée à la méthode du groupe du complément. Ainsi, on pourra le consulter et éventuellement le modifier au sein de la méthode du groupe du complément.

Avec l'objet *ObjetGV* disponible, on peut d'abord en extraire les paramètres nous permettant de savoir si le verbe est sous la forme modale ou non, s'il est sous la forme affirmative ou négative, ou s'il est à l'impératif. On doit aussi extraire de l'objet *ObjetGV* la position de verbe conjugué (possiblement un auxiliaire pour les temps composés) et la position de l'infinitif dans le cas modal. Ces informations nous permettent ensuite de déterminer quelle doit être la position du pronom-objet au sein du groupe du verbe, comme on l'a vu avec les exemples cités plus haut.

On doit ensuite pouvoir insérer le pronom-objet dans le groupe du verbe à la position qu'on vient de déterminer. Pour y arriver, on appelle la méthode « *AjouteMot* », illustrée à l'extrait de code 4.38. Cette méthode accepte en entrée les vecteurs de mots, lemmes, classes grammaticales et étiquettes morpho-syntaxiques du groupe du verbe original et du pronom à rajouter. Deux paramètres additionnels sont fournis. Le premier fournit la position où insérer le pronom (*index*). Le deuxième paramètre sert dans des cas où on doit insérer des mots après un mot précis dans un ensemble de mots de départ, ce qui n'est toutefois pas le cas ici.

Extrait de code 4.38 : Méthode « *AjouteMot* » pour ajouter le pronom-objet au groupe du verbe à la bonne position

```
public static ObjetMotLemmePOS AjouteMot(String [] mot1, String [] lemme1, int[]
                    POS1, String[] POSd1,
                    String [] mot2, String [] lemme2, int[] POS2, String[] POSd2,
                    int index, String apres){
// Insérer mot2 dans mot1 à un index donné de mot1, ou après mot précis dans mot1

ObjetMotLemmePOS objetMot= new ObjetMotLemmePOS();
String [] mot, lemme, POSd;
int [] POS;
int ctr, taille1, taille2, taille3;

// Assigner taille des arrays
taille1=mot1.length; taille2=mot2.length;
taille3=taille1+taille2;

// Trouver l'index du lemme apres, si fourni
if(!apres.isEmpty()){ // lemme apres fourni
    for(int i=0;i<taille1;i++){
        if(apres.equals(lemme1[i])) index=i;}
} // fin du if apres pas vide
if(index>(taille1-1))index=-1; // Index doit être plus petit que taille1

mot=new String[taille3]; lemme=new String[taille3];
POS=new int[taille3]; POSd=new String[taille3];

if(index==-1){ // concaténation simple array2 après array1
    for(int i=0;i<taille1;i++){
       mot[i]=mot1[i];lemme[i]=lemme1[i];POS[i]=POS1[i]; POSd[i]=POSd1[i];}
    for(int i=0;i<taille2;i++){
       mot[i+taille1]=mot2[i];lemme[i+taille1]=lemme2[i];POS[i+taille1]=POS2[i];
POSd[i+taille1]=POSd2[i];}
}
else{
    ctr=0; index--;
    for(int i=0;i<=index;i++){
      mot[i]=mot1[i];lemme[i]=lemme1[i];POS[i]=POS1[i]; POSd[i]=POSd1[i]; ctr++;}
      // jusqu'à index de array1
    for(int i=0;i<taille2;i++){
      mot[ctr]=mot2[i];lemme[ctr]=lemme2[i];POS[ctr]=POS2[i]; POSd[ctr]=POSd2[i];
ctr++;} // remplir avec array2
    for(int i=index+1;i<taille1;i++){
      mot[ctr]=mot1[i];lemme[ctr]=lemme1[i];POS[ctr]=POS1[i]; POSd[ctr]=POSd1[i];
ctr++;} // remplir avec fin de array1
} // fin du else
objetMot.SetObjetMot(mot, lemme, POS, POSd);
return objetMot;
}
```

En sortie de la méthode « *AjouteMot* », on reçoit un objet modifié (« *objetMot* ») pouvant ensuite être fourni à la méthode de type « *Set* » du groupe du verbe, pour le modifier. L'extrait de code 4.39 montre un appel typique à la méthode « *AjouteMot* » au sein du groupe du complément pour ajouter un pronom-objet au groupe du verbe. Cet appel est suivi d'un appel à la méthode « *SetMots* » pour la mise à jour de l'objet du groupe du verbe avec le pronom inséré.

Extrait de code 4.39 : Appel à la méthode « *AjouteMot* » pour ajouter le pronom-objet au groupe du verbe à la bonne position

```
verbeTemp=Ut.AjouteMot(verbeTemp.mot, verbeTemp.lemme, verbeTemp.POS,
           verbeTemp.POSd, motTemp, lemmeTemp, POSTemp, POSdTemp, position, "");

// Ensuite, modifier le Groupe Verbe avec ces changements
objetGV.SetMots("", verbeTemp.mot, verbeTemp.lemme, verbeTemp.POS,
        verbeTemp.POSd);
```

4.10.5.4.2. L'accord du participe passé conjugué avec avoir, situé avant le verbe

On a vu à la section précédente qu'il faut insérer le pronom-objet au sein du groupe du verbe, à une position précise dépendant de certaines caractéristiques du groupe du verbe. Mais il faut tenir compte d'une règle bien connue de la langue française, discutée précédemment à la Section 3.5.3.3.3 :

> Le participe passé employé avec l'auxiliaire avoir s'accorde en genre et en nombre avec le complément d'objet direct *si celui est situé devant le verbe*.

Par exemple, on pourrait avoir :

- Il a mangé les pommes. Il les a mang**ées**.

On voit qu'il ne suffit pas de positionner le pronom à la bonne position au sein du groupe du verbe. Il faut aussi s'assurer, dans le cas de temps composés, que la bonne forme du participe passé soit employée. Il faut en effet s'assurer que le participe passé conjugué avec « avoir » s'accorde en genre et en nombre avec le complément d'objet direct. Dans l'exemple tout juste cité, le pronom « les » remplace « les pommes », qui sont au féminin pluriel. Il faut donc que le participe passé soit lui aussi au féminin pluriel (« mangées »).

Par défaut, le participe passé du groupe du verbe est au masculin singulier. Pour le modifier, on commence par supprimer cette version par défaut. L'objet du groupe du verbe fournit en sortie la position du participe passé. On appelle donc la méthode « *DeleteMot* » qui envoie en entrée l'objet du groupe du verbe original, ainsi que la position du mot à supprimer. En retour, on obtient l'objet avec le mot supprimé. On appelle ensuite la même méthode « *AjouteMot* » utilisée précédemment (Section 4.10.5.4.1). Cette fois, on insère, au même endroit, la version du participe passé s'accordant correctement en genre et en nombre. La méthode « *AjouteMot* » a été introduite précédemment à l'extrait de code 4.38. La méthode « *DeleteMot* » est quant à elle fournie à l'extrait de code 4.40. En sortie de la méthode « *DeleteMot* », tout comme pour la méthode « *AjouteMot* » on reçoit un objet modifié (*objetMot*) pouvant ensuite être fourni à la méthode de type « *Set* » du groupe du verbe, pour le modifier. À l'aide des appels successifs à « *DeleteMot* » et « *AjouteMot* », on réussit donc à remplacer le participe passé par défaut (masculin singulier) par le participe passé au genre et au nombre désirés.

Extrait de code 4.40 : Méthode « *DeleteMot* » pour supprimer le participe passé original du groupe du verbe

```
public static ObjetMotLemmePOS DeleteMot(String [] motIn, String [] lemmeIn,
            int[] POSIn, String [] POSdIn, String lemmeDelete, int index){

    ObjetMotLemmePOS objetMot= new ObjetMotLemmePOS();
    String [] mot, lemme, POSd;
    int [] POS;
    int ctr, taille2;

    // Trouver l'index du lemme à enlever
    int taille=motIn.length;
    if(!lemmeDelete.isEmpty()){
       for(int i=0;i<taille;i++) if(lemmeDelete.equals(motIn[i]))index=i;
    } // fin du if lemmeDelete pour chercher l'index du mot recherché

    if(index==-1){
        mot=motIn; lemme=lemmeIn; POS=POSIn; POSd=POSdIn;} // pas de changement
    else{ // Enlever un mot selon l'index fourni ou trouvé
        ctr=0; taille2=taille-1;
        mot=new String[taille2]; lemme=new String[taille2];
        POS=new int[taille2]; POSd=new String[taille2];
        for(int i=0;i<taille;i++){
          if(index!=i){ // ajouter l'élément
              mot[ctr]=motIn[i];lemme[ctr]=lemmeIn[i];POS[ctr]=POSIn[i];
              POSd[ctr]=POSdIn[i]; ctr++;}
        } // fin du for i pour taille
    }
    objetMot.SetObjetMot(mot, lemme, POS, POSd);
    return objetMot;
}
```

4.10.5.5. Cas transitif indirect (sans infinitif)

Le cas transitif indirect s'apparente beaucoup au cas du transitif direct, à la différence qu'on introduit le groupe du nom par une préposition. On a par exemple :

- Il parle ***à*** son ami.

Lorsqu'on utilise deux noms communs, il faut se rappeler d'introduire la préposition devant les deux, en plus de la conjonction « et » :

- Il parle ***à*** son ami **et** ***à*** sa mère.

Tel que discuté à la Section 3.5.3.3.4, pour ce projet, on ne déplace jamais de pronoms au sein du groupe du verbe, dans le cas du transitif indirect, par convenance et simplicité. Par exemple, bien que tout à fait correcte, la phrase suivante ne sera pas créée par l'algorithme :

- Il parle à sa mère. Il ***lui*** parle.

La version moins élégante « Il parle à elle » sera au contraire permise, par simplicité. Donc, à l'exception de la présence des prépositions et un traitement plus simple des pronoms, le transitif indirect est traité de la même façon que le transitif direct. Aucun détail additionnel quant à l'algorithme n'est donc fourni ici.

4.10.5.6. Ajout de compléments du nom au groupe du complément

Tout comme on l'a fait (Section 4.10.4.5) au groupe du sujet, il est possible d'ajouter des compléments du nom au groupe du complément. Il faut évidemment que celui-ci comprenne un ou des noms communs. Cela exclut donc le cas de verbes intransitifs (ne demandant pas de complément) et les cas où un pronom a été inséré, plutôt qu'un nom commun. Mais une fois que ces conditions sont satisfaites, on peut en effet inclure des compléments du nom, exactement de la même façon qu'on l'a fait pour le groupe du sujet. Les méthodes utilisées et les appels à ces méthodes étant les mêmes, l'ajout de complément du nom au groupe du complément n'est pas discuté davantage, pour éviter toute répétition inutile.

4.10.5.7. Détermination de l'objet *ObjetGC*

Finalement, une fois tous les vecteurs du groupe du complément formés (incluant de potentiels compléments du nom), on doit préparer l'information complète qui doit être fournie en sortie du groupe du complément. Comme il ne reste alors aucun autre groupe à créer, l'information fournie ici n'est utile que pour bâtir la phrase complète. Comme une méthode ne peut fournir qu'un seul élément en sortie (commande *return*), on crée un objet (classe « *ObjetGC »*) contenant toute cette information, et c'est cet objet qui est retourné par la méthode « *GroupeComplement »*. Cette classe est illustrée à l'extrait de code 4.41.

Cette classe comprend un constructeur qui initialise les vecteurs selon la dimension requise. Elle comprend ensuite une méthode de type « *Set* » qui sert à assigner toutes les valeurs à toutes les variables de la classe (« *SetGC »*). Cette classe ne comprend pas de méthode de type « *Get* ». Les variables de cette classe publique seront donc extraites directement par des références du type « *ObjetGC.nomdevariable »* lorsque requises au sein des autres méthodes.

Extrait de code 4.41 : Classe « *ObjetGC »*, fournissant l'objet à fournir en sortie de la méthode du groupe du complément. Elle comprend un constructeur et une méthode de type « *Set* »

```
public class ObjetGC {

int nMotsGC, nCoocGC; String texteGC;
String[] motsGC, lemmesGC, coocGC, POSdGC;
int [] POSGC; ObjetGV objetGVGC; boolean avecNomGC;
int nombreGC=-1; int genreGC=-1;

public ObjetGC(int nMots, int nCooc){ // CONSTRUCTEUR

nMotsGC=nMots; nCoocGC=nCooc;
motsGC   = new String[nMotsGC];   lemmesGC = new String[nMotsGC];
POSdGC   = new String[nMotsGC];   coocGC   = new String[nCooc]  ;
POSGC    = new int   [nMotsGC];
}

public void SetGC(String texte,  String [] mots,  String [] lemmes, int [] POS,
                  String [] POSd,ObjetGV objetGV, String [] cooc,
                  boolean avecNom, int nombre, int genre){
texteGC=texte;
motsGC = mots; lemmesGC=lemmes; coocGC=cooc;
POSGC=POS; POSdGC=POSd; objetGVGC=objetGV;
avecNomGC=avecNom; nombreGC=nombre; genreGC=genre;
}
}
```

4.10.6. Construction des phrases

Aux sections précédentes (Sections 4.10.3, 4.10.4 et 4.10.5), les algorithmes pour créer les trois groupes de la phrase ont été présentés. En sortie de chacun de ces trois algorithmes, un objet fournit toute l'information nécessaire pour ultimement arrimer les trois groupes et générer une phrase complète. La présente section montre comment cette étape de la formation d'une phrase est effectuée.

4.10.6.1. Détermination de l'objet *ObjetPhrase*

En premier lieu, on montre à l'extrait de code 4.42 la classe « *ObjetPhrase* » dont le but est de contenir toute l'information pertinente pour la phrase complète. Au moment de bâtir le texte complet, on extrait donc l'information de cet objet pour chaque phrase. La classe « *ObjetPhrase* » est simple. Elle ne contient que six variables importantes :

- La liste de mots (vecteur de *String*)
- La liste de lemmes (vecteur de *String*)
- La liste de classes grammaticales (vecteur d'entiers)
- La liste d'étiquettes morpho-syntaxiques (vecteurs de *String*)
- Le nombre de mots de la phrase (entier)
- Le texte associé à la phrase (*String*)

La classe autrement ne contient qu'une méthode, de type « *Set* ». Elle ne sert donc qu'à emmagasiner l'information qu'on lui fournit pour chaque phrase. Cette classe ne comprend pas de méthode de type « *Get* ». Les variables de cette classe publique seront donc extraites directement par des références du type « *ObjetPhrase.nomdevariable* » lorsque requises au moment de créer le texte au complet.

Extrait de code 4.42 : Classe « *ObjetPhrase* », contenant toute l'information requise pour chaque phrase une fois formulée

```
public class ObjetPhrase {

String [] mots;
String [] lemmes;
int    [] POS;
String [] POSd;
int       nMots;
String    texte;

public void SetPhrase(String [] motsEntree, String[] lemmesEntree,
int[]POSEntree, String [] POSdEntree, String texteEntree){

mots       = motsEntree;
lemmes     = lemmesEntree;
POS        = POSEntree;
POSd       = POSdEntree;
texte      = texteEntree;
nMots      = mots.length;

}
}
```

4.10.6.2. Méthode pour la création des phrases

L'objet « *ObjetPhrase* » présenté à la section précédente nous trace la voie à suivre pour créer une phrase, dans le sens qu'on y voit l'information qu'il est nécessaire de générer pour chacune d'elles. La méthode « *PhraseStandard* » fournit justement en sortie (*return*) un objet du type « *ObjetPhrase* ». La méthode « *PhraseStandard* » crée les phrases en appelant les méthodes pour les trois groupes de chaque phrase, en combinant leur information, puis en procédant à un certain « nettoyage ». Comme cette méthode est plutôt longue, elle sera présentée plus bas, étape par étape. À l'extrait de code 4.43, on présente le début de la méthode, qui concerne la définition des variables et objets nécessaires pour la génération d'objets de type phrase. On y remarque la création d'objets en lien avec les compléments du nom de type « que », autant pour le groupe du sujet, que pour le groupe du complément (la lettre « C » ajoutée dans ce cas).

À l'extrait de code 4.44, on poursuit avec les appels aux méthodes distinctes menant à la création du groupe du verbe, du groupe du sujet ainsi que du groupe du complément, dans cet ordre précis. La séquence de ces appels est critique, puisque l'information doit circuler d'abord du groupe du verbe au groupe du sujet, puis au groupe du complément, selon les algorithmes définis pour ce projet. On remarque aussi que des appels aux méthodes pour le complément du nom sont inclus, à la fois pour le groupe du sujet et pour le groupe du complément. Le fonctionnement de ces méthodes avait été discuté à la Section 4.10.4.5.

À l'extrait de code 4.45, on combine chaque groupe de la phrase créé à l'extrait de code 4.44, pour obtenir un seul objet global pour la phrase complète. Pour y arriver, on utilise la méthode « *AjouteMot* » présentée précédemment à l'extrait de code 4.38, qui prouve ici sa versatilité.

À l'extrait de code 4.46, une fois tous les groupes de la phrase réunis, on peut procéder à divers traitements qu'on ne pouvait effectuer auparavant, puisqu'ils impliquent souvent des éléments de plus d'un groupe. Par exemple, on effectue à ce moment le déplacement de virgules au besoin, l'inclusion d'apostrophes pour les mots demandant l'élision, la contraction d'articles (« de le » devient « du ») et finalement le remplacement de certains possessifs devant des voyelles (« ma amie » devient « mon amie »). Davantage de détails sont fournis plus bas en lien avec l'incorporation d'apostrophes et la contraction d'articles.

Finalement, à l'extrait de code 4.47, on bâtit le texte de la phrase (variable de type *String*), et on crée l'objet phrase lui-même, qu'on retourne en sortie de la méthode *PhraseStandard*.

Extrait de code 4.43 : Définition des variables et objets relatifs à la phrase (début de la méthode *PhraseStandard*)

```
public ObjetPhrase PhraseStandard(boolean avecCooc){

String [] coocVide = new String[0];
String textePhrase; boolean cNom;
ObjetMotLemmePOS objetMot;
ObjetPhrase objetPhrase         = new ObjetPhrase();
ObjetMotLemmePOS objetMotCNQue  = new ObjetMotLemmePOS ();
ObjetMotLemmePOS objetMotCNQueC = new ObjetMotLemmePOS ();

(…)
```

Extrait de code 4.44 : Extrait de la méthode pour créer une phrase, qui appellera à tour de rôle les méthodes pour le groupe du verbe, le groupe du sujet et le groupe du complément

```
// Groupe Verbe principal ----------------------------------------------------
String verbeImpose="";
ObjetGV objetGV = GroupeVerbe(verbeImpose, -1, -1, -1, 0, coocVide);
CooC.global = objetGV.coocGV;

//  Groupe Sujet ------------------------------------------------------------
ObjetGS objetGS = GroupeSujet(objetGV.genreGV, objetGV.nombreGV,
                  objetGV.impersonnelGV, objetGV.subjonctifGV,
                  objetGV.tempsGV, objetGV.imperatifGV, objetGV.negatifGV,
                  objetGV.auxAvoirGV, objetGV.personneGV, CooC.global);
CooC.global = objetGS.coocGS;

// Complément du nom de type "du", "de la", ou "des" pour le SUJET -----------
cNom = !objetGV.imperatifGV;     // Condition: pas à l'impératif
cNom = cNom & objetGS.avecNomGS; // Condition: Noms dans le groupe sujet
cNom = cNom & Math.random()<Parametres.probSujetDu; // Contribution du hasard
ObjetGS objetCNDu=CompNomDu(cNom, objetGV, CooC.global);
CooC.global=objetCNDu.coocGS;

// Complément du nom de type "que", "qui", "dont", "auquel" pour le SUJET ----
cNom = !objetGV.imperatifGV;     // Condition: pas à l'impératif
cNom = cNom & objetGS.avecNomGS; // Condition: Noms dans le groupe sujet
cNom = cNom & Math.random()<Parametres.probSujetQue; // Contribution du hasard
if(cNom) objetMotCNQue = CompNomQue(objetGS.genreGS, objetGS.nombreGS);
else     objetMotCNQue.SetObjetMotVide();

// Groupe Complément ---------------------------------------------------------
ObjetGC objetGC = GroupeComplement(objetGS.genreGS, objetGS.nombreGS,
                  objetGV.attributifGV, objetGV.intransitifGV,
                  objetGV.prepositionGV, objetGV.tempsGV, objetGV.negatifGV,
                  objetGV.personneGV, objetGV, CooC.global);
CooC.global = objetGC.coocGC;

// Complément du nom de type "du", "de la", ou "des" pour le COMPLEMENT ------
cNom = !objetGV.intransitifGV;    // Condition: pas un verbe intransitif
cNom = cNom & objetGC.avecNomGC;  // Condition: Noms dans le groupe sujet
cNom = cNom & Math.random()<Parametres.probCompDu; // Contribution du hasard
ObjetGS objetCNDuComp=CompNomDu(cNom, objetGV, CooC.global);
CooC.global = objetCNDuComp.coocGS;

// Complément du nom de type "que", "qui", "dont", "auquel" pour le COMPLÉMENT --
cNom = !objetGV.imperatifGV;     // Condition: pas à l'impératif
cNom = cNom & objetGC.avecNomGC; // Condition: Noms pour le groupe sujet
cNom = cNom & Math.random()<Parametres.probCompQue; // Contribution du hasard
if(cNom) objetMotCNQueC = CompNomQue(objetGC.genreGC, objetGC.nombreGC);
else     objetMotCNQueC.SetObjetMotVide();
```

Extrait de code 4.45 : Extrait de la méthode pour créer une phrase, qui combine les différentes portions de la phrase pour créer un objet final complet pour chaque phrase

```
// Combiner les différentes sections -------------------------------------

// Combiner Groupe Sujet avec Groupe Complément du Nom de type "du"
objetMot=Ut.AjouteMot(objetGS.motsGS, objetGS.lemmesGS,   objetGS.POSGS,
                      objetGS.POSdGS, objetCNDu.motsGS, objetCNDu.lemmesGS,
                      objetCNDu.POSGS, objetCNDu.POSdGS, -1, "");

// Combiner avec Complément du Nom de type "que/qui" pour le Groupe Sujet
objetMot=Ut.AjouteMot(objetMot.mot, objetMot.lemme, objetMot.POS, objetMot.POSd,
                      objetMotCNQue.mot, objetMotCNQue.lemme, objetMotCNQue.POS,
                      objetMotCNQue.POSd, -1, "");

// Combiner avec Groupe Verbe
objetMot=Ut.AjouteMot(objetMot.mot, objetMot.lemme, objetMot.POS, objetMot.POSd,
                      objetGV.motsGV, objetGV.lemmesGV, objetGV.POSGV,
                      objetGV.POSdGV, -1, "");

// Combiner avec Groupe Complément
objetMot=Ut.AjouteMot(objetMot.mot, objetMot.lemme, objetMot.POS, objetMot.POSd,
                      objetGC.motsGC, objetGC.lemmesGC, objetGC.POSGC,
                      objetGC.POSdGC,-1, "");

// Combiner avec le complément du nom pour Groupe Complément
objetMot=Ut.AjouteMot(objetMot.mot, objetMot.lemme, objetMot.POS, objetMot.POSd,
                      objetCNDuComp.motsGS, objetCNDuComp.lemmesGS,
                      objetCNDuComp.POSGS, objetCNDuComp.POSdGS, -1, "");

// Combiner avec Complément du Nom de type "que/qui" pour le Complément
objetMot=Ut.AjouteMot(objetMot.mot, objetMot.lemme, objetMot.POS, objetMot.POSd,
                      objetMotCNQueC.mot, objetMotCNQueC.lemme,
                      objetMotCNQueC.POS, objetMotCNQueC.POSd, -1, "");
```

Extrait de code 4.46 : Extrait de la méthode pour créer une phrase, qui effectue certains traitements comme le déplacement de virgules, l'inclusion d'apostrophes, la contraction d'articles et le remplacement de certains possessifs devant des voyelles

```
// Traitements divers ----------------------------------------------------
objetMot=DeplacerVirgule(objetMot);    // Replacer les virgules au bon endroit
objetMot=Apostrophes(objetMot);        // Traitement des apostrophes
objetMot=ArticlesContractes(objetMot); // Traitement des articles contractés
objetMot=MaTaSa(objetMot);             // Remplacement devant voyelles et h
```

Extrait de code 4.47 : Extrait de la méthode pour créer une phrase, qui bâtit le texte de la phrase ainsi que l'objet phrase lui-même, fourni en sortie

```
//  Bâtir le texte de la phrase (enlever espaces inutiles, majuscules, point)
textePhrase = Ut.BatirTexte(objetMot.mot);
textePhrase = Ut.NettoyerGroupe(textePhrase);
textePhrase = Ut.MajusculeDebut(textePhrase);
if(objetGV.imperatifGV)textePhrase=textePhrase+"!";
else textePhrase=textePhrase+".";
textePhrase=Ut.TraitDunion(textePhrase); // Ajout des traits d'union
//  Bâtir l'objet phrase ------------------------------------------------
objetPhrase.SetPhrase(objetMot.mot, objetMot.lemme, objetMot.POS, objetMot.POSd,
    textePhrase, 0);
return objetPhrase;
} // fin de la méthode PhraseStandard
```

La méthode « *Apostrophes* » appelée à l'extrait de code 4.46 doit en premier lieu identifier la présence dans la phrase d'un nombre limité de mots qui peuvent devoir se transformer avec l'apostrophe (« ce », « de », « je », « jusque », etc.). Quand un tel mot est trouvé, la méthode vérifie ensuite si le mot suivant débute par une voyelle ou par un « h-muet ». Il est à noter que certains de ces mots ne demandent une apostrophe que devant les voyelles, et non devant le « h-muet ». Il existe aussi des cas particuliers. Par exemple, le mot « presque » ne demande jamais l'apostrophe, sauf devant le mot « île » (singulier ou pluriel). Le mot « quelque » ne demande jamais l'apostrophe nom plus, sauf devant les mots « un » et « une ». Finalement, le mot « si » ne demande une apostrophe que devant les mots « il » et « ils ». La méthode prend donc bien soin de vérifier toutes ces conditions. Si une apostrophe est nécessaire, alors la fonction « *substring* » élimine la dernière lettre du mot, pour qu'on puisse ensuite ajouter l'apostrophe. L'espace entre l'apostrophe tout juste ajoutée et le mot qui suit sera enlevé à l'étape subséquente.

Pour ce qui est des articles contractés, la plupart sont issus de l'union de la préposition accompagnant le verbe demandant un complément indirect avec l'article qui la suit. La méthode « *ArticlesContractes* » doit donc identifier ces quelques cas, somme toute assez peu nombreux. L'extrait de code 4.48 fournit quelques énoncés Java servant à repérer ces articles contractés, et à fournir le texte à remplacer. La variable « *m1* » contient le premier des deux mots à contracter, la variable « *m2* » contient le second. La variable « *contracte* » donne la version contractée des deux mots, tandis que les variables « *genre* » et « *nombre* » doivent être fournies dans chaque cas pour insérer les étiquettes morpho-syntaxiques des articles contractés dans l'objet de la phrase. Le détail du fonctionnement de la méthode « *ArticlesContractes* » n'est pas fourni, car plutôt trivial.

Extrait de code 4.48 : Énoncés servant à identifier les articles contractés devant être introduits dans chaque phrase, au sein de la méthode « *ArticlesContractes* »

```
m1[0]="de";m2[0]="le";  contracte[0]="du";  genre[0]=0; nombre[0]=0;
m1[1]="de";m2[1]="les"; contracte[1]="des"; genre[1]=2; nombre[1]=1;
m1[2]="à"; m2[2]="le";  contracte[2]="au";  genre[2]=0; nombre[2]=0;
m1[3]="à"; m2[3]="les"; contracte[3]="aux"; genre[3]=2; nombre[3]=1;
```

4.10.7. Construction du texte final et fichiers associés

La première étape de la construction du texte final et de ses objets associés implique de faire rouler la méthode pour la création de phrases de façon répétée, puis en enregistrant les objets pour chaque phrase dans un vecteur les contenant tous (le vecteur « *toutesPhrases* »). Les plus importantes portions de la méthode pour y arriver (« *GenererTexte* ») sont fournies à l'extrait de code 4.49.

La méthode « *GenererTexte* », comme on peut le voir, sert aussi à bâtir l'histogramme du nombre de mots par phrase. On peut ainsi voir non seulement quelle est en moyenne la longueur de chaque phrase, mais aussi la distribution de ces longueurs de phrases, pour ensuite pouvoir comparer avec les statistiques équivalentes pour le corpus de référence. Des exemples de tels histogrammes sont fournis au Chapitre 5 portant sur les résultats de ce projet.

L'étape suivante pour bâtir le texte final consiste à combiner toutes ces phrases en un seul objet, ce qui se fait avec la méthode « *CombinePhrases* » qui accepte en entrée le vecteur « *toutesPhrases* » discuté plus haut. L'extrait de code 4.50 montre la portion de cette méthode où une boucle est effectuée autant de fois qu'on a de phrases. À chaque itération, on extrait l'information de la phrase courante. L'appel à la fonction « *Concatener* » permet ensuite de joindre la phrase courante aux phrases déjà jointes, pour former éventuellement l'objet unique pour le texte en entier. En parallèle, le texte final (variable unique de type *String Builder*) se bâtit en joignant le texte de chacune des phrases.

Des méthodes sont ensuite exécutées pour afficher les statistiques globales pour le texte généré aléatoirement, ainsi que le détail pour chacun des mots. Ces méthodes ne sont pas décrites ici. Les résultats seront discutés au Chapitre 5.

Extrait de code 4.49 : Méthode servant à bâtir le texte aléatoire au complet, par l'appel répété à la méthode pour bâtir les phrases

```
public ObjetPhrase [] GenererTexte(boolean avecCooc){

Parametres.etapePhrase=true;
nPhrasesHasard=Parametres.nPhrases;
ObjetPhrase [] toutesPhrases = new ObjetPhrase[Parametres.nPhrases];

// Ajuster les fréquences de temps et de personne
ratioPersonnesA  = Ut.AjusterFreqPersonne(personneArray);
ratioPersonnesNA = Ut.AjusterFreqPersonne(personneArrayNA);


for(int i=0;i<Parametres.nPhrases;i++){
    toutesPhrases[i]=PhraseStandard(true);
    histHasard[toutesPhrases[i].nMots]++;
} // fin de toutes les phrases

// Batir histogramme du nombre de phrases
int max=0; int totalmots=0;
for(int i=0;i<100;i++){
    if (histHasard[i]!=0)max=i;
}
return toutesPhrases;
}
```

Extrait de code 4.50 : Extrait de la méthode servant à combiner toutes les phrases pour un obtenir un seul set d'objets pour le texte en entier

```
for(int i=1;i<taille;i++){
    ObjetPhrase phraseCourante=objetPhrase[i];
    String[] motCourant   = phraseCourante.mots;
    String[] lemmeCourant = phraseCourante.lemmes;
    String[] POSdCourant  = phraseCourante.POSd;
    int[] POSCourant      = phraseCourante.POS;
    String texteCourant   = phraseCourante.texte;
    ObjetMotLemmePOS objCourant = new ObjetMotLemmePOS();
    objCourant.SetObjetMot(motCourant, lemmeCourant, POSCourant, POSdCourant);
    objFinal=Ut.Concatener(objFinal, objCourant);
    texteB.append(" ");
    texteB.append(texteCourant);
    tailleCourante=motCourant.length;
    for(int j=0;j<tailleCourante;j++){
        noPhrase[nMotsTexteHasard]=i;
        nMotsTexteHasard++;
    }
}
```

4.10.8. Algorithme pour la création d'une « salade de mots »

L'algorithme pour créer la « salade de mots » décrite à la Section 3.5.6 utilise les mêmes mots que ceux des phrases automatiquement lemmatisées au hasard. Mais ces mots sont ici « brassés » et donc insérés dans un ordre complètement aléatoire. La série de mots qui en résulte est ensuite groupée en « phrases » individuelles, qu'on fait commencer par une majuscule et terminer par un point. Mais ces phrases ne respectent aucune syntaxe ni sémantique. Les longueurs des phrases de la salade de mots sont choisies afin de reproduire les longueurs des phrases générées au hasard à l'étape précédente.

Cette salade de mots se retrouve donc à avoir exactement le même nombre de phrases et de mots que le texte aléatoire automatiquement lemmatisé généré précédemment, ainsi qu'exactement les mêmes mots. Les mots de la salade de mots sont aussi associés aux mêmes lemmes et étiquettes morpho-syntaxiques. Il en résulte donc que les statistiques globales pour la salade de mots sont identiques à celles pour le texte généré au hasard à l'étape précédente. L'extrait de code 4.51 illustre la méthode pour créer la salade de mots.

En entrée, on envoie l'objet unique final du texte aléatoire final (« *objetPhrase* »). C'est de cet objet que pratiquement toute l'information est extraite pour créer la salade. Mais on envoie aussi en entrée le vecteur comprenant toutes les phrases individuelles (« *toutesPhrases* »). Ce vecteur ne sert qu'à extraire la longueur de chacune des phrases, pour reproduire ces mêmes longueurs de phrases pour la salade de mots. Après l'initialisation des variables et objets, on appelle la méthode « *OrdreEntier* », qui ne fait que mettre dans une séquence aléatoire tous les nombres entiers entre zéro et le nombre total de mots du texte généré au hasard. C'est donc dans l'ordre aléatoire dicté par cette méthode que les mots vont se glisser dans la salade de mots. Une boucle est donc exécutée pour assigner tous les mots, accompagnés de leurs lemmes, classes grammaticales et étiquettes morpho-syntaxiques, dans la salade de mots, selon cette séquence.

Par la suite, un nettoyage du texte a lieu. En particulier, on enlève toutes les apostrophes qui étaient présentes initialement, car dans la plupart des cas, elles ne sont plus nécessaires, le mot en élision n'est plus nécessairement suivi par un mot débutant par une voyelle ou un « h-muet ».

Mais il faut par la suite réintroduire des apostrophes là où elles deviennent nécessaires, selon la nouvelle séquence de mots. Après avoir identifié le début et la fin de chaque « phrase » de la salade, on s'assure que chacune débute par une majuscule et se termine par un point. Comme étape finale, on bâtit l'objet complet de la salade de mots (« *texteSalade »*), comme on l'avait fait précédemment pour les phrases générées au hasard, incluant la variable *String* finale pour la salade de mots en tant que tel (*texte*).

Pour bien illustrer le concept de « salade de mots » développé par l'algorithme de l'extrait de code 4.51, le Tableau 4.23 donne l'exemple d'un très court texte de deux phrases dans sa version originale ainsi que dans une version de type « salade de mots ».

À l'exemple du Tableau 4.23, on constate que les mêmes mots ont été utilisés des deux côtés. On remarque que le « l' » devant le mot « accotement » s'est retrouvé devant le mot « à » qui débute lui aussi par une voyelle. On remarque aussi que deux phrases ont été formées, tout comme dans le texte original, et que le nombre de mots par phrase (8 et 5) a été reproduit dans la même séquence pour la salade de mots.

Des exemples de salades de mots générées pour ce projet sont fournis au Chapitre 5 et à l'Annexe I.

Tableau 4.23 : Exemple d'un très court texte dans sa version originale et dans une version de type « salade de mots »

Texte original	Exemple de salade de mots possible pour ce texte
La voiture amorce son virage à grande vitesse. Elle empiète sur l'accotement.	Vitesse la sur amorce grande à voiture l'elle. Empiète accotement virage son.

Extrait de code 4.51 : Méthode pour créer la salade de mots

```
public ObjetPhrase Salade(ObjetPhrase objetPhrase, ObjetPhrase[] toutesPhrases){

ObjetPhrase texteSalade = new ObjetPhrase();
String texte;
int ctrMot=0; int ctrPhrase=0;
StringBuilder texteB = new StringBuilder("");

String [] motIn   = objetPhrase.mots;
String [] lemmeIn = objetPhrase.lemmes;
int    [] POSIn   = objetPhrase.POS;
String [] POSdIn  = objetPhrase.POSd;
int taille = motIn.length;
String [] motFinal   = new String[taille];
String [] lemmeFinal = new String[taille];
int    [] POSFinal   = new int   [taille];
String [] POSdFinal  = new String[taille];

// Déterminer le nouvel ordre au hasard des mots
int [] ordreRevise =Ut.OrdreEntier(taille);

// Extraire le nombre de mots par phrases
int nPhrases=toutesPhrases.length;
int [] lp = new int[nPhrases]; // Longueur des phrases
for(int i=0;i<nPhrases;i++)lp[i]=toutesPhrases[i].nMots;

for(int i=0;i<taille;i++){
    ctrMot++;
    // Mettre mots dans le nouvel ordre
    motFinal  [i] = motIn  [ordreRevise[i]];
    lemmeFinal[i] = lemmeIn[ordreRevise[i]];
    POSFinal  [i] = POSIn  [ordreRevise[i]];
    POSdFinal [i] = POSdIn [ordreRevise[i]];
    String prochainMot;
    if(i!=taille-1) prochainMot = motIn [ordreRevise[i+1]];
    else prochainMot="";
    if(motFinal[i].equals("cet"))motFinal[i]="ce";
    // Enlever toutes les apostrophes
    if(motFinal[i]!=null && motFinal[i].contains("'"))motFinal[i]=lemmeFinal[i];

    // Vérifier début de phrase - mettre majuscule
    String motInclure;
    if(ctrMot==1) motInclure=Ut.MajusculeDebut(motFinal[i]);
    else          motInclure=motFinal[i];
    // Vérifier fin de phrase - pour compteur à zéro et apostrophes
    String finPhrase=" ";
    if(ctrMot==lp[ctrPhrase]){ // Fin de la phrase atteint
        ctrMot=0; ctrPhrase++; // Mettre compteurs à jour
        finPhrase=". ";        // Rajouter un point à la fin
    }
    else{ // Pas fin de phrase, donc tester apostrophe
        motInclure=ApostropheSalade(motInclure,prochainMot);
        if(motInclure.contains("'"))finPhrase="";
    }
    texteB.append(motInclure); texteB.append(finPhrase);
}
texte=texteB.toString();
texteSalade.SetPhrase(motFinal, lemmeFinal, POSFinal, POSdFinal, texte, 1);
return texteSalade;
}
```

5. RÉSULTATS OBTENUS

Maintenant que la méthodologie et les algorithmes Java en lien avec la lemmatisation, la désambiguïsation d'homographes et la génération de phrases automatiquement lemmatisées ont été décrits aux Chapitres 3 et 4, on peut enchainer avec les résultats obtenus lors de leur exécution. Cette section est séparée en deux grandes parties : l'une portant sur l'Étape 1 (Sections 5.1 à 5.5), donc principalement sur la lemmatisation et l'analyse du corpus de référence, l'autre portant sur l'Étape 2 (Section 5.6), donc la génération de textes aléatoires automatiquement lemmatisés.

5.1. Lemmatisation de base du corpus

L'Étape 1 de ce projet consiste à générer l'information nécessaire pour alimenter les algorithmes de l'Étape 2 qui eux servent par la suite à générer des phrases aléatoires automatiquement lemmatisées. L'Étape 1 inclut les étapes principales listées ici :

- Lecture des banques de données lexicales de base
- Création de vecteurs, matrices, tables de hachage et objets pour emmagasiner les banques de données lexicales de base
- Génération des verbes conjugués selon les listes de verbes et tableaux de conjugaison
- Lecture et préparation du corpus de référence
- Lemmatisation de base du corpus de référence
- Désambiguïsation des homographes
- Génération des cooccurrences
- Génération des statistiques globales du corpus de référence
- Préparation des informations à fournir pour l'Étape 2

Les algorithmes en lien avec la lecture des données lexicales de base et le chargement du corpus de référence ont été suffisamment discutés au chapitre précédent. L'exécution de ces algorithmes mène à la création de bon nombre de variables et tables de hachage servant ensuite de matériel de base à l'exécution d'algorithmes plus avancés dont l'analyse s'avère davantage pertinente. Ainsi, plutôt que de s'attarder aux éléments produits en sortie par les algorithmes de base, la section présente débute avec les résultats fournis par la lemmatisation de base du corpus de référence. Tel qu'on l'a mentionné précédemment, la lemmatisation de base, telle qu'on la définit ici, exclut toute opération de désambiguïsation des homographes. Celle-ci s'effectue à une étape ultérieure. Toujours est-il que des statistiques intéressantes sur la quantité d'homographes présents dans le corpus peuvent être extraites lors de la lemmatisation de base. Ces statistiques s'avèrent très pertinentes car elles témoignent de l'ampleur de la tâche à venir, considérant la grande proportion d'homographes contenue dans le corpus de référence, et par extension, dans tout texte de langue française. Ainsi, à la Section 5.1.1, on expose ces statistiques portant sur les homographes, alors qu'aux Sections 5.1.2 et 5.1.3, on s'attardera aux proportions de classes grammaticales et de temps et de personnes de verbes. Ces proportions influencent de façon importante l'efficacité des algorithmes de désambiguïsation, discutés par la suite à la Section 5.2.

5.1.1. Homographes présents

Tel que mentionné au Tableau 3.12 du Chapitre 3, on retrouve en général des homographes impliquant des lemmes appartenant à diverses classes grammaticales. On cherche donc d'abord à confirmer cette observation dans le contexte d'un des corpus référence utilisés pour ce projet, le roman « Le Rouge et le Noir ». Au Tableau 5.1, on fait un décompte de toutes les paires de classes grammaticales observées pour les homographes de ce corpus. Par exemple, pour l'homographe « demande », le compteur correspondant à « verbe vs. nom » est incrémenté. Il est

à noter que les homographes impliquant plus de deux classes grammaticales affectent davantage d'éléments du Tableau 5.1. Par exemple, la présence de l'homographe « ferme » dans le corpus fait incrémenter les compteurs correspondant à « nom vs. adjectif », « nom vs. verbe », et « verbe vs. adjectif ». La même approche s'applique aux homographes impliquant davantage de classes grammaticales, comme le mot « tout » qui peut appartenir, selon le contexte, à l'une de cinq classes différentes. Les classes grammaticales sont extraites en comptabilisant les classes qu'on retrouve pour chaque mot du corpus dans la table de hachage « *tableRef* » qui avait été illustrée à la Figure 4.1.

Au Tableau 5.1, on comptabilise le nombre d'homographes *distincts* impliquant chaque paire de classes grammaticales. Autrement dit, on ne s'occupe pas de la fréquence d'apparition de chaque homographe. Ainsi, chaque homographe n'est ici considéré qu'une seule fois. Le Tableau 5.1 est symétrique, puisque par exemple, chaque homographe de type « adjectif vs. verbe » est aussi de type « verbe vs. adjectif ». La diagonale du tableau (cases foncées) correspond aux homographes issus d'une même classe grammaticale, comme par exemple le mot « suis », associé aux verbes « être » ou « suivre ». Le Tableau 5.1 nous indique que les paires de classes les plus fréquentes sont de loin :

- Adjectifs vs. verbes (total de 1827 cas distincts, pourcentage de 52%)
- Noms vs. verbes (total de 947 cas distincts, pourcentage de 27%)
- Noms vs. adjectifs (total de 639 cas distincts, pourcentage de 18%)

Ces trois formes représentent donc à elles seules 97% des formes *distinctes* d'homographes. Il faut toutefois interpréter les résultats du Tableau 5.1 avec prudence, car on se rappelle que les homographes associés à plus d'une classe grammaticale (par exemple « ferme » ou « tout ») y sont comptabilisés plus d'une fois. Une analyse plus précise nous indique que le corpus compte en fait 2896 homographes distincts. Les homographes issus de la même classe grammaticale (somme des éléments de la diagonale = 86) comptent donc pour 3.0% de tous les homographes distincts recensés dans le corpus.

Tableau 5.1 : Quantité d'homographes distincts dans le roman « Le Rouge et le Noir », en fonction des classes grammaticales impliquées

	Verbes	Adjectifs	Noms	Adverbes	Déterminants	Pronoms	Prépositions	Conjonctions	Interjections
Verbes	68	1827	947	4	0	2	5	2	1
Adjectifs	1827	5	639	13	10	5	1	1	0
Noms	947	639	7	15	4	2	7	2	3
Adverbes	4	13	15	1	2	2	0	1	0
Déterminants	0	10	4	2	0	18	2	0	0
Pronoms	2	5	2	2	18	5	0	3	0
Prépositions	5	1	7	0	2	0	0	0	0
Conjonctions	2	1	2	1	0	3	0	0	0
Interjections	1	0	3	0	0	0	0	0	0

Le Tableau 5.2 fournit lui aussi des statistiques sur les paires de classes grammaticales pour les homographes. Mais cette fois, les statistiques sont basées sur les *fréquences d'apparition* de chaque homographe dans le corpus, plutôt que de ne considérer que les formes distinctes. Ainsi, si on retrouve par exemple le mot « demande » 50 fois dans le corpus, alors l'élément « verbe vs. nom » sera incrémenté 50 fois pour ce mot. On recense en tout 58 866 homographes, ce qui représente 32% des mots du corpus. C'est donc dire que près d'un mot sur trois dans le corpus de référence est un homographe, ce qui est considérable.

Tout comme le Tableau 5.1, le Tableau 5.2 est symétrique et sa diagonale (cases foncées) correspond aux homographes issus de la même classe. Les homographes issus de la même classe grammaticale représentent d'ailleurs 15% de tous les homographes du corpus, en considérant les fréquences. Aussi, partout où on observait des zéros au Tableau 5.1, on observe aussi forcément des zéros au Tableau 5.2. Il est intéressant de se concentrer à nouveau sur les valeurs les plus élevées du tableau. On note cette fois :

- Pronoms vs. déterminants (18 607 occurrences, pourcentage de 24%)
- Déterminant vs. préposition (11 169 occurrences, pourcentage de 14%)
- Noms vs. verbes (10 691 occurrences, pourcentage de 14%)

Ces trois catégories à elles seules comptent pour 51% du total. En comparant les Tableaux 5.1 et 5.2, on note que bien que le cas « pronom vs. déterminant » n'implique que 18 homographes distincts (par exemple « le », « la », pour une proportion totale de 0.51%), ceux-ci représentent une bien plus grande proportion de tous les homographes, lorsqu'on considère leur fréquence d'apparition (24%). À l'inverse, la paire « adjectifs vs. verbes », qui représente 52% de tous les homographes distincts, ne représente que 11% des homographes, quand on considère leur fréquence d'apparition.

Tableau 5.2 : Quantité d'homographes dans le roman « Le Rouge et le Noir », en fonction des classes grammaticales impliquées

	Verbes	Adjectifs	Noms	Adverbes	Déterminants	Pronoms	Prépositions	Conjonctions	Interjections
Verbes	981	8708	10691	1385	0	10	420	10	27
Adjectifs	8708	80	7328	2592	851	1002	1	7	0
Noms	10691	7328	590	4119	2196	820	404	122	29
Adverbes	1385	2592	4119	754	1508	1508	0	767	0
Déterminants	0	851	2196	1508	0	18607	11169	0	0
Pronoms	10	1002	820	1508	18607	6172	0	4868	0
Prépositions	420	1	404	0	11169	0	0	0	0
Conjonctions	10	7	122	767	0	4868	0	0	0
Interjections	27	0	29	0	0	0	0	0	0

Le Tableau 5.3 fournit quant à lui la liste des 20 homographes les plus courants du roman « Le Rouge et le Noir ». Aucun de ces mots n'est véritablement caractéristique du corpus de référence, puisqu'on les retrouve de façon générale en grande proportion dans tout document de langue française. On peut en déduire que peu importe le corpus de référence choisi, on risque de se retrouver avec une liste d'homographes assez semblable lorsqu'on s'attarde aux plus fréquents.

Finalement, la Figure 5.1 reproduit le résultat du Tableau 5.3 de façon graphique. Superposé au graphique à barres pour les fréquences des homographes les plus fréquents dans le corpus, cette figure inclut aussi la courbe de la loi de Zipf. Cette loi dicte que de façon générale, la fréquence d'utilisation d'un mot dans un texte volumineux est inversement proportionnelle à son rang. Bien que l'accord entre les données « expérimentales » en bleu pour le corpus de référence et la courbe de la loi de Zipf en orange ne soit pas parfait, on constate tout de même que la tendance est bel et bien respectée.

Tableau 5.3 : Liste des homographes les plus fréquents dans le roman « Le Rouge et le Noir », avec leurs fréquences, lemmes et classes respectifs

Mot	Fréquence	Lemmes et classes grammaticales
de	9047	de (déterminant et préposition)
la	4078	le (déterminant et pronom)
le	3854	le (déterminant et pronom)
l'	3317	le (déterminant et pronom)
que	2305	que (pronom et conjonction)
d'	2122	de (déterminant et préposition)
les	1978	le (déterminant et pronom)
qu'	1613	que (pronom et conjonction)
est	1283	être (verbe), est (nom)
son	1232	son (nom et déterminant)
pas	1189	pas (nom et adverbe)
plus	1081	plaire (verbe), plus (adverbe)
s'	950	si (conjonction), se (pronom)
dit	781	dire (verbe), dit(adjectif*)
si	767	si (conjonction, adverbe)
tout	754	tout (adjectif, nom, adverbe, déterminant et pronom)
bien	596	bien (nom et adverbe)
être	410	être (verbe et nom)
même	383	même (adjectif et adverbe)
fait	359	faire (verbe), fait (adjectif*, nom)

** Les mots « dit » et « fait » sont considérés comme des adjectifs car dans ce projet, tous les participes passés sont aussi considérés comme des adjectifs. Voir Section 4.1.4*

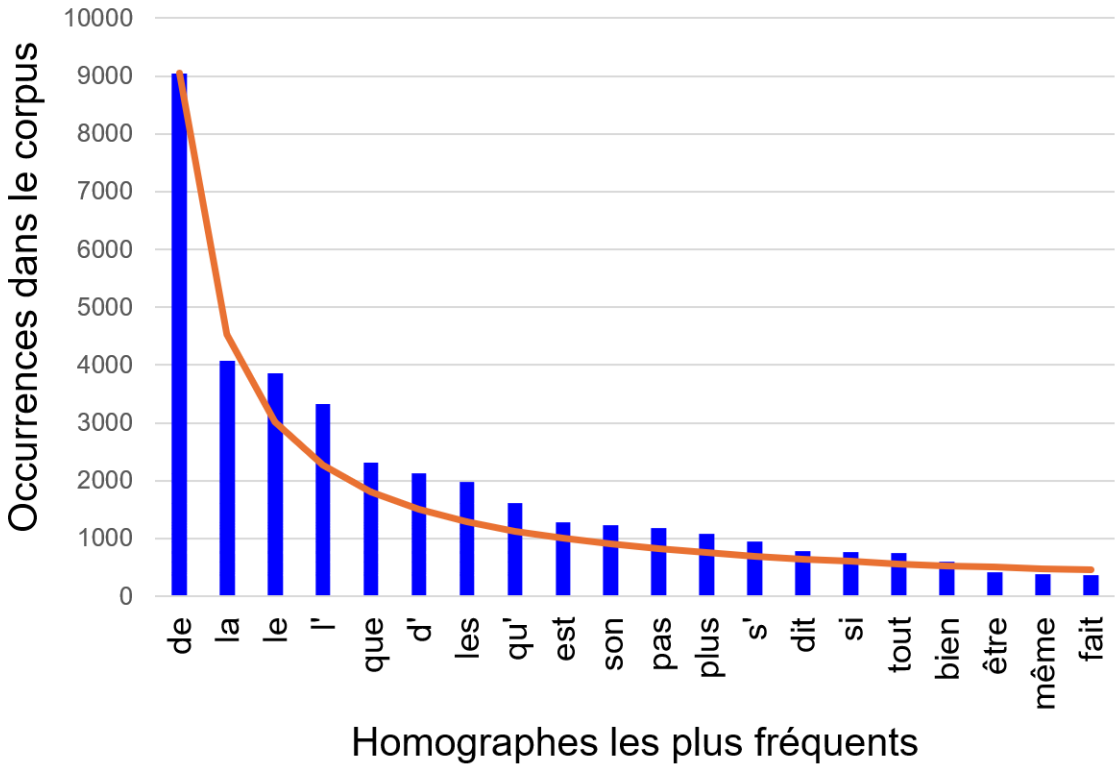


Figure 5.1 : Homographes les plus fréquents dans le roman « Le Rouge et le Noir ».
La courbe orange représente la loi de Zipf

5.1.2. Proportions des classes grammaticales

L'objectif de l'algorithme de désambiguïsation des homographes dans le projet actuel est d'associer chaque homographe à son lemme et à sa classe grammaticale appropriés. Comme on le verra à la Section 5.2, la désambiguïsation sera facilitée par l'identification d'indices d'appartenance de chaque mot à une classe grammaticale. Le Tableau 5.4 présente les proportions de classes grammaticales dans le roman « Le Rouge et le Noir », en fonction des fréquences totales observées pour tous les mots (pas uniquement les homographes). Cependant, comme ces données sont fournies avant que la désambiguïsation des homographes ait eu lieu, on ne peut encore déterminer à quelle classe grammaticale appartiennent les homographes pouvant appartenir à plus d'une classe, ce qui est le cas pour la grande majorité d'entre eux. Le Tableau 5.4 présente donc les proportions de classes grammaticales selon deux calculs. Dans le premier cas, on exclut tous les homographes, ne considérant donc que les mots non ambigus. Dans le deuxième cas, on n'exclut aucun mot, mais à chaque fois que l'on rencontre un homographe, on incrémente le compte pour chacune des classes grammaticales auxquelles le mot peut en théorie appartenir. Ainsi par exemple, le mot « demande » fera augmenter à la fois le nombre de verbes et le nombre de noms. Il en résulte que lorsqu'on fait la somme des pourcentages pour les neuf classes grammaticales, on obtient une valeur inférieure à 100% dans le premier cas, puisque nous avons exclu un grand nombre de mots (les homographes). La situation inverse est observée dans le deuxième cas où les homographes sont comptabilisés plus d'une fois, ce qui entraine un pourcentage total supérieur à 100%. L'intérêt de cette approche est qu'on peut être assuré que les proportions *véritables*, celles qu'on obtiendrait à la suite d'une désambiguïsation *parfaite* des homographes, doivent se retrouver à quelque part entre les deux valeurs extrêmes affichées au Tableau 5.4.

Tableau 5.4: Proportions des neuf classes grammaticales selon les fréquences dans le roman « Le Rouge et le Noir » avant désambiguïsation (en incluant et en excluant les homographes)

Classe grammaticale	En excluant les homographes	En assignant toutes les classes possibles de chaque homographe
Verbes	13%	23%
Adjectifs	3.5%	11%
Noms	14%	25%
Adverbes	4.6%	7.7%
Déterminants	8.8%	24%
Pronoms	11%	22%
Prépositions	7.0%	13%
Conjonctions	3.1%	6.2%
Interjections	0.10%	0.13%
TOTAL	65%	131%

Ainsi, la proportion véritable de verbes doit se retrouver à quelque part entre 13% et 23%, celle des adjectifs entre 3.5% et 11%, et ainsi de suite. On constate que les différences de valeurs entre les deux colonnes vont environ du simple au double, ce qui témoigne de la présence imposante d'homographes dans le roman, et par extension, dans tout texte de langue française. On revisitera ces proportions une fois les homographes désambiguïsés (Section 5.3.5.3), pour confirmer que les proportions se retrouveront en effet entre les limites listées au Tableau 5.4.

5.1.3. Proportions des personnes et temps de verbes

On se rappelle qu'un des objectifs du projet actuel est de générer des textes aléatoires automatiquement lemmatisés. Aussi, on souhaite que ces textes suivent le « modèle » du corpus de référence dont ils s'inspirent, ce qui veut dire qu'on tente d'y introduire les personnes de verbe (1ère, 2e, et 3e personnes du singulier et du pluriel) et les temps de verbe (présent, imparfait, etc.) dans les mêmes proportions que pour le corpus de référence. On cherche donc à déterminer ces proportions au sein du corpus de référence. Le Tableau 5.5 affiche les proportions pour les personnes de verbe, alors que le Tableau 5.6 affiche les proportions pour les temps de verbe. On procède ici de la même façon que pour les classes grammaticales, en présentant les résultats en deux colonnes. La première colonne ne considère que les cas non ambigus, c'est-à-dire ceux pour lesquels il n'y a qu'une seule possibilité pour la personne ou pour le temps du verbe. À la deuxième colonne, on inclut toutes les possibilités, c'est-à-dire qu'on comptabilise toutes les possibilités de personnes et de temps de verbe pour la forme verbale courante. Par exemple, pour la forme verbale « aime », il y a trois possibilités de personnes : 1ère comme dans « j'aime », 3e comme dans « il aime » et 2e comme dans l'impératif « aime! ». Dans la même veine, le verbe « aime » peut se retrouver à trois temps différents, soit à l'indicatif présent, au subjonctif présent, ainsi qu'à l'impératif. Tout comme pour le Tableau 5.4, on se retrouve donc avec des totaux inférieurs à 100% pour la première colonne (on exclut les cas ambigus) et supérieurs à 100% pour la deuxième colonne (les cas ambigus sont associés à plus d'un cas). Il faut toutefois noter que lorsqu'il est question d'un verbe à l'infinitif, ou au participe présent ou participe passé, on n'assigne pas de personne du verbe. Ces formes verbales sont donc exclues du calcul du pourcentage des personnes.

Tableau 5.5: Proportions des personnes de verbe selon les fréquences dans le roman « Le Rouge et le Noir » avant désambiguïsation (en incluant et en excluant les cas ambigus)

Personnes de verbe	En excluant les cas ambigus (possibilité de plus d'une personne de verbe)	En assignant toutes les personnes possibles à chaque forme verbale
1ère du singulier	2.3%	30%
2e du singulier	0.17%	31%
3e du singulier	42%	74%
1ère du pluriel	0.19%	0.92%
2e du pluriel	1.3%	3.2%
3e du pluriel	4.2%	6.1%
TOTAL	50%	146%

On constate que la différence entre les valeurs des deux colonnes du Tableau 5.5 est encore plus grande que ce qu'on l'on avait observé avec les classes grammaticales au Tableau 5.4. Cela est dû au très grand nombre de formes verbales ambiguës en français, comme on l'a vu avec l'exemple « aime ». Là encore, on peut s'attendre à ce que les véritables proportions se retrouvent à quelque part entre les valeurs des deux colonnes. Cependant, il faut noter que le projet actuel cherche uniquement à désambiguïser les homographes selon leur classe grammaticale. Aucun algorithme n'a été mis en place pour tenter de déterminer avec certitude la personne du verbe en cas d'ambiguïté. On ne peut donc pas s'attendre à un bien meilleur estimé de ces proportions à la suite des opérations de désambiguïsation des homographes décrites plus loin.

Le Tableau 5.6 fournit des données dans un format similaire, mais cette fois pour les temps de verbe. Là encore, on constate d'énormes écarts entre les valeurs des deux colonnes, tout particulièrement pour le présent et pour le subjonctif présent, puisque énormément de formes verbales (celles au premier groupe) sont identiques pour ces deux temps aux premières et troisièmes personnes (« j'aime », « que j'aime », « il aime », « qu'il aime »). À l'impératif, on dénote aussi un écart énorme. Il appert que pour l'impératif, les cas non ambigus sont extrêmement rares. On retrouve par exemple « sache », « sachons » et « sachez » du verbe savoir, des formes qu'on ne retrouve à aucun autre temps qu'à l'impératif. Mais très peu d'autres exemples sont possibles, ce qui explique la valeur de près de 0% pour l'impératif au Tableau 5.6. Il est à noter qu'aucun temps composé (passé composé, plus-que-parfait, etc.) n'est listé ici, les participes passés ayant tous été classifiés comme tel, sans chercher à savoir s'ils étaient associés à un auxiliaire particulier, et le temps de cet auxiliaire. Il faut aussi se rappeler que tous les participes passés dans ce travail sont aussi par défaut considérés comme des adjectifs (participes passés employés seuls). Ce qui explique la valeur nulle de proportion de participes passés pour le cas non ambigu.

Tout comme pour les personnes de verbe, la désambiguïsation des homographes n'apportera pas beaucoup d'informations additionnelles pour déterminer les véritables proportions, car les algorithmes développés ici ne cherchent pas à déterminer les temps de verbe dans les cas ambigus. Mais la désambiguïsation des homographes améliorera tout de même quelque peu ces résultats. En effet, certains homographes pouvant en théorie être des verbes peuvent dans les faits appartenir à une autre classe grammaticale dans le contexte du corpus de référence. Ainsi, on s'est retrouvé aux Tableaux 5.5 et 5.6 à déterminer la personne et le temps pour certains mots qui en réalité, ne sont même pas des verbes dans le contexte où ils sont utilisés.

Tableau 5.6: Proportions des temps de verbe selon les fréquences dans le roman « Le Rouge et le Noir » avant désambiguïsation (en incluant et en excluant les cas ambigus)

Temps de verbe	En excluant les cas ambigus (possibilité autre classe, ou plus d'une personne de verbe)	En assignant toutes les personnes possibles à chaque forme verbale
Infinitif	12%	15%
Présent	5.0%	33%
Imparfait	15%	16%
Passé simple	10%	16%
Futur simple	2.2%	2.4%
Subjonctif présent	0.22%	16%
Subjonctif imparfait	1.1%	1.2%
Impératif	0.012%	16%
Conditionnel	1.1%	1.5%
Participe présent	1.6%	3.4%
Passé composé	0.0%	18%
TOTAL	48%	140%

5.2. Désambiguïsation des homographes : entrainement

Tel qu'on l'a démontré à la Section 5.1.1, près du tiers des mots du roman « Le Rouge et le Noir » sont des homographes. Cette proportion considérable donne une idée de la taille du défi de désambiguïsation. On se rappelle que la désambiguïsation des homographes, dans le projet actuel, permet une meilleure identification des fréquences des lemmes employés au sein du corpus de référence, ce qui en retour permettra de générer des phrases aléatoires à l'Étape 2 du projet en lien étroit avec le corpus.

Tel que discuté à la Section 3.2.3, seuls les homographes impliquant des classes grammaticales différentes sont désambiguïsés pour ce projet. Comme on l'a mentionné à la Section 5.1.1, seule une faible proportion des homographes n'implique qu'une seule classe grammaticale. C'est donc dire que les algorithmes développés pour ce projet s'attaquent à la grande majorité des homographes observés dans le corpus de référence. La stratégie pour désambiguïser ces homographes de classes grammaticales distinctes est de miser sur les indices syntaxiques dans la phrase. Cela inclut la position relative des mots de différentes classes, ainsi que l'accord en genre, nombre et personnes des différents mots composant chaque phrase. Les indices utilisés, appelés ici « caractéristiques » ont été présentés à la Section 4.6.1. Un algorithme d'apprentissage machine, ici la régression logistique binaire, a donc été développé sur la base de ces caractéristiques. L'apprentissage machine s'exécute en deux étapes principales, la première étant l'entrainement, sujet de la section courante, la deuxième étant l'évaluation, qui sera traitée à la Section 5.3.

Deux approches distinctes ont été adoptées pour l'entrainement de l'apprentissage machine pour ce projet, soit l'approche automatique et l'approche manuelle. Ces deux approches, discutées en détail aux Sections 4.6.3.1 et 4.6.3.2, comportent chacune leurs avantages et inconvénients. Aux sections suivantes (5.2.1 et 5.2.2), on discute des mots du corpus servant à nourrir l'algorithme

d'entrainement. Les résultats en sortie de l'étape d'entrainement consistent en des fichiers de caractéristiques qui serviront pour l'évaluation. Ces fichiers de caractéristiques seront discutés à la Section 5.2.3.

5.2.1. Entrainement automatique – ensemble d'entrainement

A la Section 4.6.1, on mentionnait que l'entrainement dit « automatique » s'effectue en analysant les caractéristiques des mots *non ambigus*, donc non homographes. C'est-à-dire qu'on ne s'attarde qu'aux mots dont on connaît avec certitude la classe grammaticale, puisqu'il n'y en a qu'une seule possible. Par exemple, le mot « voiture » ne peut être qu'un nom, et le mot « finirais » ne peut être qu'un verbe. Cette façon de faire évite tout recours à un entrainement manuel fastidieux, d'où l'intérêt de cette approche. On notait à la Section 5.1.1 que dans le roman « Le Rouge et le Noir » par exemple, 32% des mots sont des homographes. Le corollaire est que les mots composant le 68% restant sont forcément des mots non ambigus. L'entrainement automatique peut donc se baser sur cette proportion importante (environ les deux tiers) des mots du corpus.

Mais on se rappelle aussi qu'à la Section 4.6.3.1, on proposait une variation de l'entrainement automatique dans laquelle seul un sous-ensemble des mots non ambigus était utilisé. L'objectif de cette approche, dite « limitée », par opposition à « complète », était de minimiser l'ambiguïté des classes grammaticales des mots entourant directement l'homographe sous étude. En effet, en appliquant l'approche limitée, on n'utilise que les homographes n'ayant aucun homographe comme voisin immédiat, tel qu'illustré précédemment à la Figure 4.23.

Le Tableau 5.7 fournit le nombre total de mots composant l'ensemble d'entrainement dans le cas de l'entrainement automatique pour les deux corpus de référence utilisés pour ce projet. On y présente les données en fonction des deux approches d'entrainement automatique, soit l'approche « complète » et l'approche « limitée ». Ces mots sont aussi groupés par classe grammaticale. Ce sont ces mots qui serviront ensuite à nourrir l'algorithme d'entrainement, comme on le verra à la Section 5.2.4. On constate que l'approche limitée réduit le nombre d'éléments de l'ensemble d'entrainement par plus de la moitié. On verra à la Section 5.3.1 si cette limitation a porté fruit en ce qui concerne la performance de l'algorithme de désambiguïsation.

Tableau 5.7: Nombre de mots compris dans l'ensemble d'entrainement lors de l'entrainement automatique

	Corpus 1 : « Le Rouge et le Noir »		**Corpus 2 : « Science-Fiction »**	
Classe grammaticale	Approche complète	Approche limitée	Approche complète	Approche limitée
Verbes	23505 (19%)	10849 (18%)	21859 (19%)	9618 (19%)
Adjectifs	6347 (5.1%)	2749 (4.7%)	5772 (4.9%)	2545 (4.9%)
Noms	31941 (26%)	12108 (21%)	30220 (26%)	10714 (21%)
Adverbes	8320 (6.7%)	4631 (7.8%)	9431 (8.0%)	4454 (8.6%)
Déterminants	16022 (13%)	8183 (14%)	13867 (12%)	6283 (12%)
Pronoms	20030 (16%)	11668 (20%)	19208 (16%)	10645 (21%)
Prépositions	12680 (10%)	5757 (9.7%)	11842 (10%)	4576 (8.8%)
Conjonctions	5592 (4.5%)	3008 (5.1%)	5251 (4.5%)	2792 (5.4%)
Interjections	183 (0.15%)	111 (0.19%)	184 (0.16%)	148 (0.29%)
Total	124620 (100%)	59064 (100%)	117634 (100%)	51775 (100%)

En comparant les pourcentages pour chaque classe grammaticale, on note d'abord que ceux-ci ne sont pas énormément affectés par l'approche adoptée (complète ou limitée). On remarque aussi que les pourcentages sont très comparables entre les deux corpus étudiés. On peut donc émettre l'hypothèse que les pourcentages du Tableau 5.7 se retrouveraient pour la plupart des textes de langue française, du moins pour des romans. Finalement, ces proportions de classes grammaticales retrouvées dans les ensembles d'entrainement se retrouvent entre les valeurs extrêmes illustrées précédemment au Tableau 5.4, ce qui veut dire que l'ensemble d'entrainement pour l'entrainement automatique est représentatif de l'ensemble du texte à ce chapitre.

5.2.2. Entrainement manuel – ensemble d'entrainement

L'entrainement manuel représente le type d'entrainement traditionnel pour l'apprentissage machine. Il consiste à classifier manuellement chaque observation considérée lors de l'entrainement. Comme on le mentionnait à la Section 4.6.3.2, l'entrainement manuel requiert d'identifier la classe grammaticale à laquelle chaque homographe appartient réellement, dans le contexte de la phrase à lemmatiser. Comme environ le tiers du corpus de référence est composé d'homographes, il en résulte que pour un long texte comportant par exemple 100 000 mots, il faudra manuellement analyser environ 33 000 d'entre eux, ce qui représente une tâche considérable. Pour minimiser cet effort, on ne s'est attardé ici qu'à une fraction du corpus de référence. On n'a donc désambiguïsé manuellement qu'une portion de tous les homographes. Il va de soi que plus le bassin d'homographes désambiguïsés manuellement est grand, plus le modèle aura le potentiel d'être efficace. Il a donc fallu choisir un compromis acceptable entre l'effort à déployer, et le niveau d'efficacité recherché.

Pour ce projet, on s'est limité à désambiguïser environ 10 000 homographes, au lieu des 60 000 (environ) faisant partie du corpus en entier. On reviendra sur ce point à la Section 5.3.2, au moment de présenter la performance de la désambiguïsation dans le cas de l'entrainement manuel. Mais comme on le mentionnait aussi au Chapitre 4, on bénéficie au moment de l'entrainement manuel d'un effet de levier. En effet, on ne dispose pas uniquement de l'information sur les 10 000 homographes désambiguïsés manuellement, on peut aussi tirer profit de la présence de tous les mots non ambigus (non homographes) dans les phrases qui ont été désambiguïsées manuellement. Ainsi, le bassin de mots formant l'ensemble d'entrainement pour l'entrainement manuel comprend ces deux sous-ensembles : les homographes désambiguïsés manuellement, ainsi que les non homographes. En moyenne donc, pour chaque homographe désambiguïsé manuellement, on ajoute deux non homographes à l'ensemble d'entrainement.

Le Tableau 5.8 dénombre les mots utilisés pour bâtir l'ensemble d'entrainement pour les deux corpus de référence, en fonction de leur classe grammaticale. On y indique aussi explicitement lesquels parmi ces mots sont des homographes qui ont été manuellement désambiguïsés. Ce sont les mots dénombrés au Tableau 5.8 qui serviront ensuite à nourrir l'algorithme d'entrainement, comme on le verra à la Section 5.2.4.

Tableau 5.8: Nombre de mots compris dans l'ensemble d'entrainement lors de l'entrainement manuel

	Corpus 1 : « Le Rouge et le Noir »		**Corpus 2 : « Science-Fiction »**	
Classe grammaticale	Nombre d'homographes	Nombre total de mots	Nombre d'homographes	Nombre total de mots
Verbes	1126 (10%)	5492 (16%)	1562 (14%)	5261 (17%)
Adjectifs	1213 (11%)	2386 (6.9%)	1175 (11%)	2080 (6.7%)
Noms	1570 (14%)	7802 (23%)	1707 (15%)	6921 (22%)
Adverbes	802 (7.3%)	2357 (6.9%)	900 (8.1%)	2460 (8.0%)
Déterminants	2712 (25%)	5709 (17%)	2784 (25%)	5110 (17%)
Pronoms	814 (7.4%)	4313 (13%)	726 (6.6%)	4040 (13%)
Prépositions	2195 (20%)	4626 (14%)	1743 (16%)	3693 (12%)
Conjonctions	587 (5.3%)	1651 (4.8%)	463 (4.2%)	1252 (4.1%)
Interjections	5 (0.0%)	33 (0.10%)	9 (0.10%)	53 (0.20%)
Total	11024 (100%)	34369 (100%)	11069 (100%)	30870 (100%)

On constate d'abord en s'attardant aux totaux du Tableau 5.8 que tel que prévu, le nombre total de mots de l'ensemble d'entrainement est environ le triple du nombre d'homographes, puisque pour un homographe, on retrouve en moyenne deux non-homographes. Cette observation est semblable pour les deux corpus utilisés.

En comparant les pourcentages pour chaque classe grammaticale au Tableau 5.8, on remarque que les pourcentages sont très comparables entre les deux corpus étudiés. On peut donc émettre l'hypothèse, comme on l'avait fait pour le Tableau 5.7 (entrainement automatique) que les pourcentages du Tableau 5.8 se répèteraient pour la plupart des textes de langue française, du moins pour des romans. Finalement, ces proportions de classes grammaticales retrouvées dans les ensembles d'entrainement se retrouvent entre les valeurs extrêmes illustrées précédemment au Tableau 5.4, ce qui veut dire que l'ensemble d'entrainement pour l'entrainement manuel est lui aussi représentatif de l'ensemble du texte à ce chapitre.

5.2.3. Tableaux de caractéristiques

Une fois que les mots de l'ensemble d'entrainement sont déterminés, que ce soit selon l'entrainement automatique ou manuel, il faut ensuite évaluer toutes les caractéristiques pour chacun des mots. Pour ce projet, un total de 65 caractéristiques a été utilisé, c'est donc dire que pour chaque mot de l'ensemble de test, l'algorithme assigne une valeur à chacune de ces 65 caractéristiques. Les 65 caractéristiques utilisées sont détaillées à l'Annexe B.

On bâtit donc un tableau comprenant autant de lignes qu'il y a de mots dans l'ensemble d'entrainement, et autant de colonnes qu'il y a de caractéristiques (65), plus un. Il faut en effet ajouter la colonne comprenant la classe grammaticale (entier compris entre 1 et 9). Par convenance, la classe grammaticale occupe la première colonne. Le Tableau 5.9 fournit quelques données du tableau correspondant à l'entrainement manuel pour le roman « Le Rouge et le Noir ». On n'y inclut que les premières lignes, ainsi que les premières et dernières colonnes, par souci de concision.

Tableau 5.9: Extrait du tableau final de caractéristiques pour le roman « Le Rouge et le Noir », pour l'entrainement manuel. Seules les premières lignes sont affichées, ainsi que les premières et dernières colonnes, par souci de concision

Classe grammaticale	Caract 1	Caract 2	Caract 3	Caract 4	Caract 5
5	0	0	0	0	0
2	0	0	0	0	1
3	0	1	0	0	0
7	0	0	1	0	0
3	0	0	0	0	0
1	0	0	1	0	0
1	1	0	0	0	0
7	1	0	0	0	0
5	0	0	0	0	0
5	0	0	0	0	1

(…)

Caract 61	Caract 62	Caract 63	Caract 64	Caract 65
0	0	0	0	0
0	0	0	0	0
0	1	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

On constate d'abord au Tableau 5.9 que les valeurs des 65 caractéristiques sont toutes égales à 0 ou à 1, où 1 indique que la caractéristique est présente (ou vraie) pour le mot en question. On constate aussi que les classes grammaticales correspondent au tout début du texte du roman « Le Rouge et le Noir ». En effet, les premiers mots du roman sont illustrés à la Figure 5.2, accompagnés du code de leur classe grammaticale. Ces codes concordent parfaitement avec les chiffres notés à la première colonne du Tableau 5.9. Dans le cas de l'entrainement automatique, seuls les non homographes seraient considérés. Dans le cas de l'approche complète, on commencerait donc plutôt par la suite « 2-3-3-1-1-7 », correspondant aux non homographes « petite », « ville », « Verrières », « peut », « passer » et « pour ».

Il est bon de mentionner ici que peu importe la méthode utilisée pour l'entrainement (approche automatique complète, approche automatique limitée, méthode manuelle), on se retrouve dans tous les cas avec un tableau semblable à celui du Tableau 5.9. C'est-à-dire un tableau comprenant 66 colonnes, dont la première colonne correspond à la classe grammaticale connue de chaque mot et dont les autres colonnes correspondent aux valeurs des caractéristiques. Le nombre de lignes varie en revanche en fonction de l'approche ou de la méthode utilisée. Ainsi, l'approche automatique limitée comportera moins de lignes que l'approche automatique complète, et le nombre de lignes pour la méthode manuelle dépendra de la portion du texte désambiguïsée manuellement. C'est donc dire qu'une fois ce tableau généré, le reste de l'algorithme d'entrainement est identique peu importe l'approche adoptée.

A la Section 4.6.5, il a été question de « tests spécialisés », où des caractéristiques spécifiques sont déterminées pour certains homographes en particulier. Ces tests spécialisés requièrent eux aussi des tableaux de caractéristiques semblables au Tableau 5.9, mais ne seront pas illustrés ici, puisque le principe est le même. La performance des tests spécialisés sera analysée à la Section 5.3.2.1.

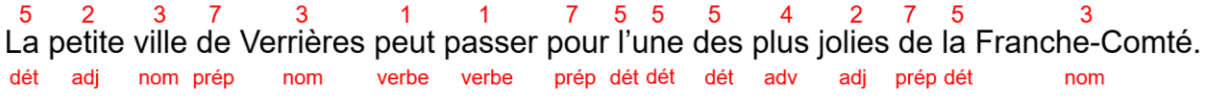


Figure 5.2 : Première phrase du roman « Le Rouge et le Noir ». Chaque mot est accompagné de sa classe grammaticale (code au-dessus et abréviation au-dessous)

5.2.4. Application de l'algorithme d'entrainement

Une fois le tableau de caractéristiques bâti (Tableau 5.9), tous les ingrédients sont disponibles pour procéder à l'entrainement en tant que tel. L'algorithme groupe d'abord les résultats selon toutes les paires possibles de classes grammaticales, pour établir des équations de régression logistique pour chacune de ces paires. Par exemple, un fichier ne contiendra que les lignes du Tableau 5.9 correspondant aux classes « 1 » et « 2 » (verbes et adjectifs). Un autre fichier ne contiendra que les lignes correspondant aux classes « 1 » et « 3 » (verbes et noms), et ainsi de suite. Tel que démontré au Tableau 4.16, il existe en tout 36 paires possibles de classes grammaticales considérant les neuf classes adoptées pour ce projet. Il y aura donc un total de 36 fichiers créés, combinant les classes grammaticales par paires. Le Tableau 5.10 fournit en exemple un extrait du fichier correspondant aux classes « 1 » et « 2 » (verbes et adjectifs) pour le roman « Le Rouge et le Noir », obtenu par entrainement manuel. On y constate en effet, que seules les valeurs « 1 » et « 2 » apparaissent à la première colonne.

L'algorithme de régression logistique binaire crée donc, pour chaque paire de classes grammaticales, un vecteur de facteurs (valeurs de type double) de dimension égale au nombre de caractéristiques, plus un. En effet, le facteur additionnel correspond à l'ordonnée à l'origine (constante) pour chaque équation de régression. Chacun des fichiers regroupant une paire de classes grammaticales peut ensuite être fourni en entrée à un algorithme de régression logistique binaire. En effet, la régression binaire s'applique ici, puisque nous ne considérons que deux classes à la fois. Un algorithme tel que la fonction « *mnrfit* » de MATLAB, ou tout algorithme semblable, peut être utilisé pour générer les vecteurs de la régression, qui serviront par la suite à prédire la classe grammaticale de chaque mot à évaluer dans le texte. L'outil de régression logistique binaire n'est pas expliqué plus en détail ici, puisqu'il s'agit d'un outil standard appliqué ici sans aucune modification.

Avec l'ensemble des vecteurs obtenus pour les 36 paires de classes grammaticales se conclut l'étape d'entrainement de l'apprentissage machine. Les prochaines sections porteront donc sur l'étape suivante et finale de l'apprentissage machine : l'évaluation.

Tableau 5.10: Extrait du tableau final de caractéristiques pour le roman « Le Rouge et le Noir », pour l'entrainement manuel, pour les classes grammaticales 1 et 2 (verbes et adjectifs). Seules les premières lignes sont affichées, ainsi que les premières et dernières colonnes, par souci de concision. En plus de celui-ci, 35 autres tableaux semblables sont générés pour toutes les paires de classes grammaticales possibles

Classe grammaticale	Caract 1	Caract 2	Caract 3	Caract 4	Caract 5
2	0	0	0	0	1
1	0	0	1	0	0
1	1	0	0	0	0
2	0	0	0	1	0
2	0	0	0	0	1
2	0	0	1	0	0
2	0	0	1	0	0
2	0	0	1	0	0
1	0	0	0	0	0
2	0	0	0	0	0

(…)

Caract 61	Caract 62	Caract 63	Caract 64	Caract 65
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	1	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

5.3. Désambiguïsation des homographes : évaluation

L'apprentissage machine comporte deux grandes étapes : l'entrainement et l'évaluation. Une fois l'entrainement effectué tel que décrit à la section précédente, on peut procéder à l'évaluation. Celle-ci s'effectue typiquement dans deux contextes distincts. D'une part, on peut vouloir tester l'efficacité de l'algorithme d'apprentissage machine. Dans un tel cas, on applique l'algorithme à un ensemble de test dont toutes les classes grammaticales sont déjà connues. On peut donc ainsi comparer les prédictions faites par l'algorithme avec les valeurs véritables. D'autre part, on peut aussi appliquer l'algorithme à un texte encore non analysé, justement pour procéder à sa lemmatisation complète dans un but autre que celui de quantifier la performance de l'algorithme lui-même. Les sections suivantes (Sections 5.3.1 à 5.3.4) se concentrent sur la quantification de la performance de l'algorithme de désambiguïsation, en comparant ses prédictions avec des valeurs connues, donc obtenues par désambiguïsation manuelle.

L'étape d'évaluation est en quelque sorte l'image miroir de l'étape d'entrainement. Plutôt que de développer un modèle à partir de données connues, on part de données inconnues (les classes grammaticales d'homographes) qu'on tente de prédire sur la base des équations développées au moment de l'entrainement. Et pour cette évaluation, on ne se concentrera cette fois que sur les homographes, puisqu'il est inutile de tenter de prédire la classe grammaticale de mots pour lesquels il n'en existe de toute façon qu'une seule possible.

Au moment de l'évaluation, l'algorithme analyse un mot à la fois en « calculant » son vecteur de caractéristiques. Autrement dit, dans le contexte du projet actuel, on assigne des valeurs (typiquement des « 0 » et des « 1 ») aux 65 caractéristiques listées à l'Annexe B pour chaque mot. Ce vecteur de caractéristiques sera alors utilisé dans les équations de régression logistique développées au moment de l'entrainement.

L'équation ou les équations utilisées dépendent des classes grammaticales théoriquement possibles pour l'homographe sous étude. Comme on le démontrait à la Section 5.1.1, en grande majorité, les homographes n'offrent que deux possibilités de classes grammaticales. Par exemple, l'homographe « demande » ne peut qu'être soit un verbe, soit un nom. Il ne peut par exemple être un adjectif ou un adverbe, ni appartenir à aucune autre classe grammaticale. Ainsi, pour désambiguïser l'homographe « demande », on fera appel à l'équation de régression logistique binaire concernant uniquement les *verbes* et les *noms*, soit une seule des 36 équations développées au moment de l'entrainement. Lors de l'application de cette équation sur la base des caractéristiques du mot, une valeur inférieure à 0.5 indique que l'algorithme prédit qu'il s'agit d'un verbe, tandis qu'une valeur supérieure à ce seuil correspond à la prédiction d'un nom. La même approche est adoptée pour tous les homographes ne comportant que deux classes possibles. Seule l'équation de régression adoptée varie, selon les classes grammaticales possibles du mot. Quand plus de deux classes grammaticales sont possibles pour un mot, on procède tel que mentionné précédemment à la Section 4.6.2.2.2, c'est-à-dire en combinant le résultat de plus d'une équation. La procédure précise ne sera pas répétée ici.

Mais plusieurs paramètres peuvent influencer la performance de l'algorithme de désambiguïsation. Par exemple, comme on l'a vu à la Section 5.2, les fichiers de caractéristiques à partir desquels on détermine les vecteurs de régression dépendent du type d'entrainement effectué (automatique ou manuel). Dans le cas de l'entrainement automatique, la performance dépendra aussi de l'approche adoptée, soit l'approche automatique complète ou l'approche automatique limitée. Pour ce qui est de l'entrainement manuel, la performance dépendra aussi d'un certain nombre de facteurs. Par exemple, la quantité d'homographes utilisés pour l'entrainement. Pour l'évaluation manuelle, certains autres facteurs seront aussi considérés, tels que l'usage de tests spécialisés. On aborde en premier lieu à la section suivante l'évaluation de la performance pour le cas de l'entrainement automatique.

5.3.1. Performance basée sur l'entrainement automatique

Tel que décrit précédemment, deux approches ont été utilisées pour l'entrainement automatique : l'approche complète et l'approche limitée, selon que les homographes entourés d'autres homographes sont considérés ou non lors de l'entrainement. Il faut se rappeler, tel qu'illustré précédemment à la Figure 4.25c, qu'il est tout à fait correct pour le cas « automatique » d'utiliser le même texte et les mêmes phrases pour à la fois l'entrainement et l'évaluation. La raison est que ce ne sont pas les mêmes mots qui sont utilisés pour l'entrainement et l'évaluation. En effet, on n'utilise que les non ambigus (non homographes) pour l'entrainement, et uniquement les homographes pour l'évaluation. Il n'y a donc ainsi aucun chevauchement entre l'ensemble d'entrainement et l'ensemble de test, même si on se base exactement sur le même texte. On pourra donc ignorer pour l'instant tout souci en lien avec la définition des ensembles de test et d'évaluation.

Dans un premier lieu, on déterminera laquelle des deux approches automatiques (complète ou limitée) s'est avérée la plus efficace, dans le cas du roman « Le Rouge et le Noir ». Pour répondre à cette question, une grande matrice de confusion comportant 9 lignes et 9 colonnes est bâtie. Ce nombre de lignes et de colonnes correspond au nombre de classes grammaticales définies dans le cadre de ce projet. Cette matrice de confusion permettra d'analyser en détail la performance de l'algorithme de désambiguïsation. Deux matrices sont affichées. À la Figure 5.3, on montre la matrice de confusion correspondant à l'approche automatique complète, tandis que la Figure 5.4 fait de même pour l'approche automatique limitée. La diagonale (cases vertes) correspond aux prédictions correctes, qu'on souhaite les plus nombreuses possible. Les cases rouges correspondent au contraire aux mauvaises prédictions. Par exemple, la valeur « 132 » affichée à la deuxième colonne de la première ligne à la Figure 5.3 nous indique qu'à 132 reprises, l'algorithme a erronément prédit qu'un verbe était un adjectif. Les cases blanches quant à elles correspondent à des cas qui ne se sont pas matérialisés. Par exemple, l'algorithme n'a jamais confondu un verbe avec un déterminant. Ces deux matrices de confusion affichent aussi dans leur marge, les valeurs de précision (horizontalement, au bas du tableau), les valeurs de rappel (verticalement, à la droite du tableau), et finalement la performance globale (dans le coin inférieur droit).

Tel qu'on l'espérait, la performance globale s'est avérée supérieure avec l'approche limitée, mais la différence est somme toute minime. En effet, la performance a passé de 81.9% à 82.2% en optant pour l'approche limitée, une hausse d'à peine 0.3%. Les mêmes opérations ont été effectuées avec le deuxième corpus de référence (le roman de science-fiction). Bien que les matrices de confusion ne soient pas affichées ici pour ce corpus, on a aussi noté une amélioration avec l'approche limitée. Mais là encore, une hausse plutôt négligeable d'à peine 0.3% a été observée, la performance globale passant de 83.4% à 83.7%. Tout de même, l'approche limitée demeure préférable, pas tellement pour sa plus grande performance, mais surtout pour le fait que l'exécution de la désambiguïsation est plus rapide, car basée sur un nombre plus restreint de données de départ.

Les deux matrices de confusion des Figures 5.3 et 5.4 nous informent aussi sur les plus grands défis de la désambiguïsation, lorsqu'on s'attarde aux plus grandes valeurs au sein des cases rouges. La plus grande valeur de toutes les cases rouges correspond au cas « préposition vs. déterminant ». On constate en effet (Figure 5.3) que 658 prépositions ont erronément été identifiées par l'algorithme comme étant des déterminants. Comme on le verra plus tard lors de la discussion des tests spécialisés (Section 5.3.2.3), il s'agit principalement d'erreurs concernant les homographes « de » et « d' ». Un autre chiffre passablement élevé correspond au cas « pronom vs. conjonction ». Dans ce cas (Figure 5.4), on dénote 273 cas où un pronom a été erronément identifié comme étant une conjonction. Il s'agit ici d'erreurs concernant les homographes « que » et « qu' ». On y reviendra aussi à la Section 5.3.2.3.

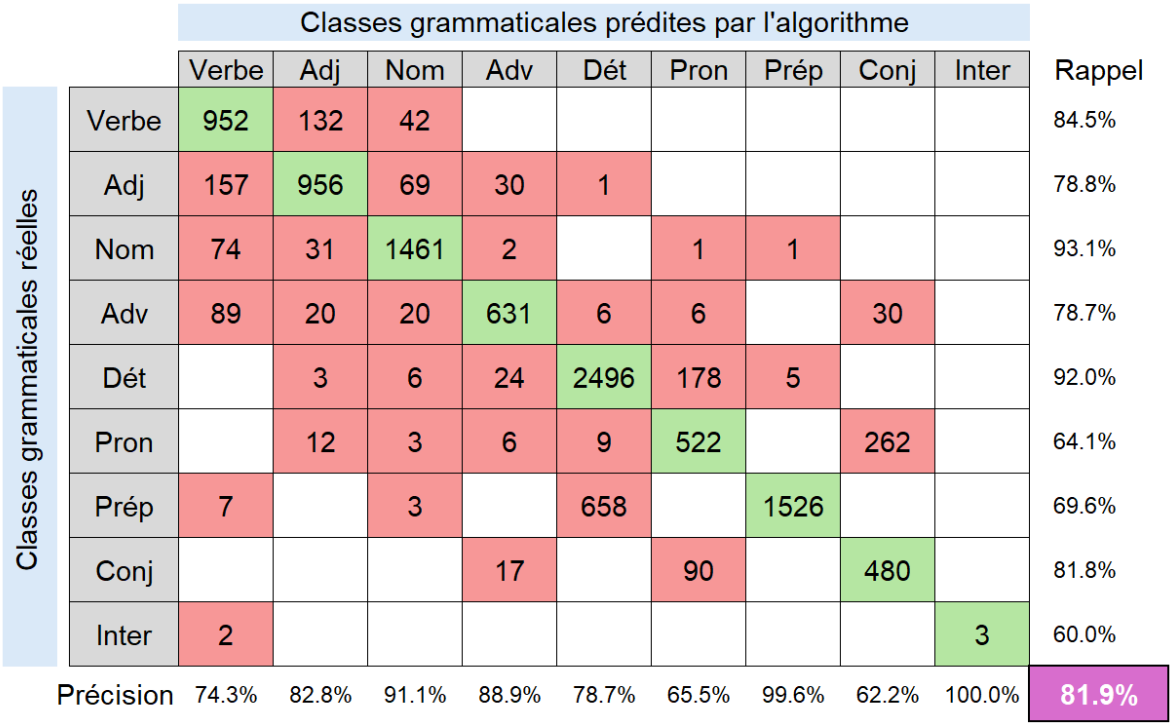


Figure 5.3 : Matrice de confusion pour l'approche automatique complète

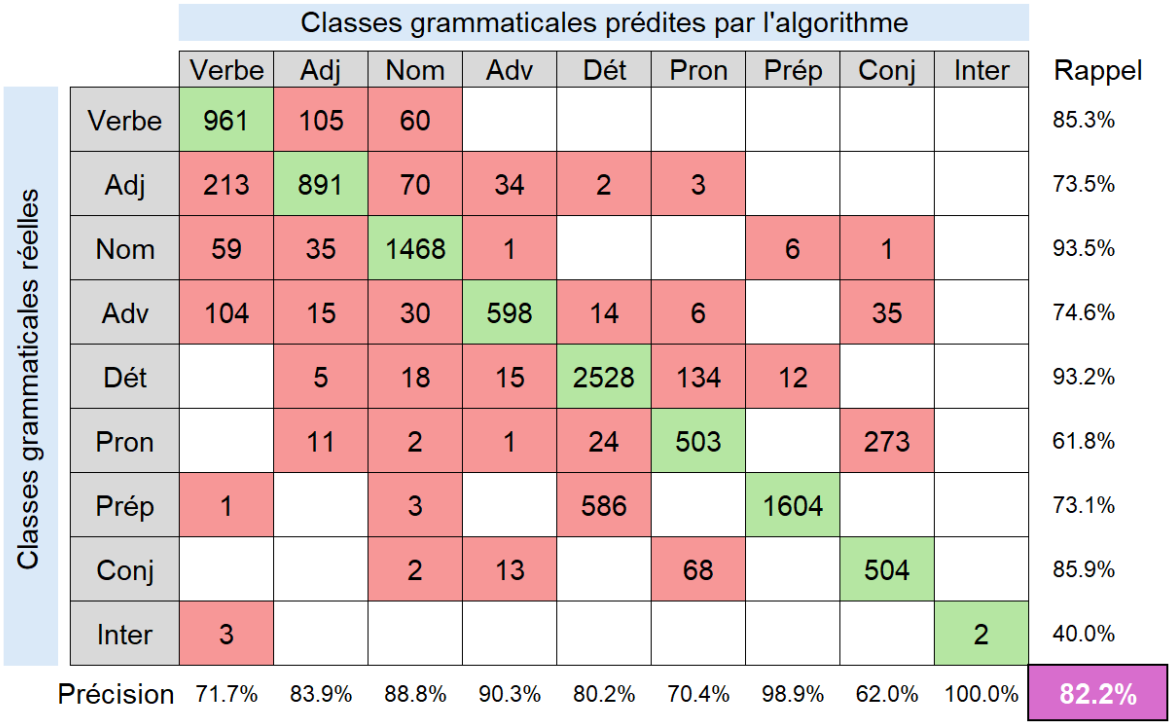


Figure 5.4 : Matrice de confusion pour l'approche automatique limitée

Tableau 5.11 : Performance globale de la désambiguïsation avec l'approche automatique limitée, en considérant les 4 combinaisons possibles d'ensembles d'entrainement et de test

		Corpus utilisé pour l'évaluation	
		Le Rouge et le Noir	Science-Fiction
Corpus utilisé pour l'entrainement	Le Rouge et le Noir	82.2%	83.4%
	Science-Fiction	81.7%	83.7%

Comme on le mentionnait plus haut, en lien avec la Figure 4.25c, il est correct d'utiliser le même texte pour l'entrainement et l'évaluation dans le cas de l'entrainement automatique, puisqu'il n'y a aucun chevauchement entre l'ensemble d'entrainement et l'ensemble de test. Mais il est tout de même pertinent de vérifier la performance de la désambiguïsation quand on applique les résultats de l'entrainement à un texte autre que celui utilisé pour l'entrainement. Comme deux corpus de référence ont été utilisés pour ce travail, on peut donc facilement explorer de tels résultats. Le Tableau 5.11 donne les résultats de la performance globale de la désambiguïsation dans quatre cas distincts, c'est-à-dire pour les quatre combinaisons possibles de textes utilisés pour l'entrainement et pour l'évaluation, considérant les deux corpus disponibles.

Au Tableau 5.11, les valeurs des cases bleues correspondent aux résultats discutés précédemment, où le même corpus est utilisé pour l'entrainement et l'évaluation. Les cases vertes quant à elles correspondent aux cas où un corpus différent est utilisé pour les deux opérations. Les résultats du Tableau 5.11 ne sont pas surprenants. On constate en effet que la performance globale de désambiguïsation est supérieure quand le même corpus est utilisé pour les deux opérations de l'apprentissage machine (entrainement et évaluation). En effet, la performance de l'évaluation du roman « Le Rouge et le Noir » baisse légèrement de 82.2% à 81.7% lorsque l'entrainement se base plutôt sur le corpus de science-fiction. De la même façon, la performance globale de l'évaluation du corpus de science-fiction baisse de 83.7% à 83.4%, quand on utilise à l'entrainement l'autre corpus, plutôt que le roman de science-fiction lui-même. Mais la bonne nouvelle est que les diminutions observées sont faibles (0.5% et 0.3% respectivement). On peut donc en conclure que l'on peut appliquer avec confiance les résultats d'entrainement sur un corpus en particulier, pour en évaluer un autre.

L'analyse de la performance de la désambiguïsation des homographes basée sur l'entrainement automatique se termine ici. On se réserve une analyse plus approfondie pour les cas plus intéressants impliquant l'évaluation manuelle. Cette analyse débute à la prochaine section.

5.3.2. Performance basée sur l'entrainement manuel

La désambiguïsation des homographes par apprentissage machine s'effectue principalement sur la base des caractéristiques grammaticales décrites à l'Annexe B. Mais d'autres outils qui avaient été décrits au Chapitre 4 contribuent aussi à la performance de l'algorithme global (Sections 4.6.5, 4.7.1, 4.7.2, et 4.7.3). Un à un, ces outils additionnels seront intégrés et on analysera leur contribution à l'amélioration de la performance globale, basée ici sur l'entrainement manuel.

Mais tout d'abord (Section 5.3.2.1), on étudiera l'effet sur la performance de désambiguïsation du nombre d'homographes utilisés lors de l'entrainement manuel. Cette question ne se posait pas dans le cas de l'entrainement automatique, car aucun homographe n'est utilisé dans ce cas et on peut ainsi tirer parti du corpus entier. Nous enchainerons ensuite avec l'effet de la sélection des ensembles d'entrainement et des ensembles de test (Section 5.3.2.2) sur la valeur globale de performance, une question qui avait été abordée à la Section 4.6.4.1.

5.3.2.1. Influence du nombre d'homographes utilisés à l'entrainement

Pour l'entrainement automatique décrit à la Section 5.2.1, il était possible de considérer le corpus en entier, puisque aucun effort manuel n'était nécessaire. En effet, l'entrainement automatique s'effectue uniquement sur la base de mots non ambigus (non homographes), donc des mots dont la classe grammaticale est unique et connue. L'entrainement manuel en revanche, requiert une désambiguïsation manuelle d'homographes, donc effectuée par un humain, pour générer l'ensemble d'entrainement. C'est sur la base de cet ensemble d'entrainement que l'algorithme de régression logistique est ensuite effectué. Mais comme cet effort manuel peut s'avérer fastidieux, seule une portion du corpus de référence a été désambiguïsée manuellement. L'objectif de la présente section est de démontrer que la portion désambiguïsée sélectionnée s'est avérée suffisamment grande pour assurer une bonne performance de l'algorithme de régression.

Pour illustrer l'influence du nombre d'homographes utilisés à l'entrainement, on teste l'efficacité de l'approche manuelle, en augmentant progressivement le nombre d'homographes désambiguïsés manuellement lors de l'entrainement. Mais il faut préciser que quelques simplifications ont été apportées à cette analyse. Tout d'abord, dans tous les cas étudiés dans la présente section, le même ensemble de test est utilisé, regroupant tous les homographes désambiguïsés manuellement pour le roman « Le Rouge et le Noir ». C'est donc dire que l'ensemble d'entrainement se retrouve inclus au sein de l'ensemble de test, et que pour le dernier cas étudié, les deux ensembles se chevauchent entièrement. Cette approche est illustrée à la Figure 5.5.

L'approche illustrée à la Figure 5.5 n'est pas l'approche normalement préconisée pour l'apprentissage machine, car on cherche généralement à éviter tout chevauchement entre les données d'entrainement et les données de test. On a choisi cette approche ici par simplicité, reconnaissant que les niveaux de performance obtenus devraient correspondre à un maximum théorique. On émet l'hypothèse ici que ce choix d'approche n'aura pas d'effet important sur la relation qu'on tente de démontrer entre le nombre d'homographes inclus dans l'ensemble d'entrainement et la performance globale. L'étude de la sélection des ensembles d'entrainement et de test sera présentée à la section suivante.

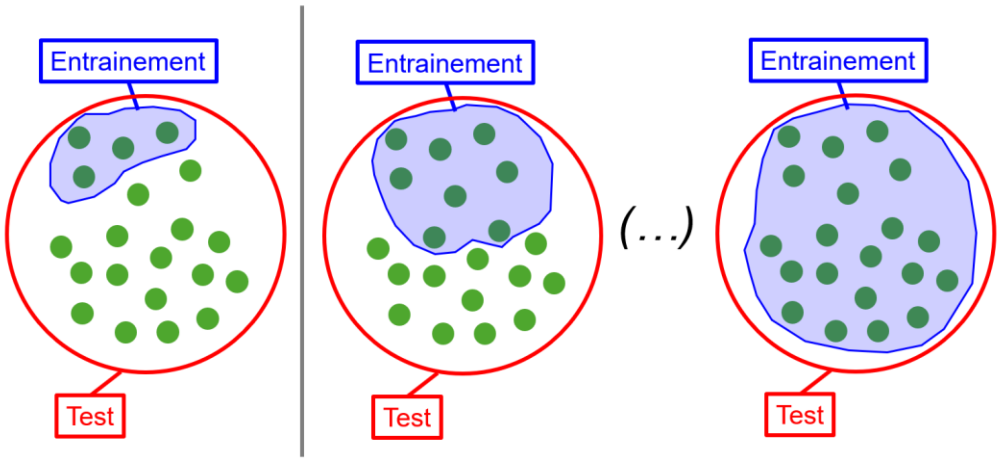


Figure 5.5 : Ensembles d'entrainement et de test sélectionnés pour évaluer l'influence du nombre d'homographes utilisés à l'entrainement. On augmente progressivement le nombre d'homographes de l'ensemble d'entrainement en conservant le même ensemble de test

Aussi, les outils additionnels augmentant la performance de la désambiguïsation, tels que les tests spécialisés, l'identification de locutions et l'analyse statistique des verbes, n'ont pas été considérés ici, encore là, par simplicité, et aussi parce que ces outils sont décrits plus en détail dans les sections à venir. Finalement, les résultats obtenus ici ont été obtenus avec l'analyse de la phrase dite « gauche-droite » (voir Section 4.6.4.3).

La Figure 5.6 illustre la performance de la désambiguïsation obtenue en fonction du nombre d'homographes utilisés à l'entrainement. On y constate que la performance est très basse lorsque peu d'homographes sont utilisés. Une performance d'à peine 47% est obtenue par exemple, en n'utilisant que 500 homographes dans l'ensemble d'entrainement. La performance augmente ensuite progressivement, pour finalement atteindre un plateau et se stabiliser au-delà de 8000 homographes, se situant autour de 87%. On en conclut pour l'instant, vu l'atteinte d'un plateau de performance, que le nombre d'homographes utilisés pour la désambiguïsation manuelle s'est avéré suffisant.

On constate aussi que la performance atteinte pour l'approche manuelle (environ 87%) est supérieure à la performance que nous avions obtenue avec l'approche automatique (environ 82% pour le même corpus). C'est donc dire que l'approche manuelle, sans surprise, donne de meilleurs résultats. On verra cependant aux sections suivantes, que cette performance manuelle s'améliorera davantage, en incorporant les autres outils discutés plus haut (tests spécialisés, analyse des participes passés et locutions, statistiques sur les verbes).

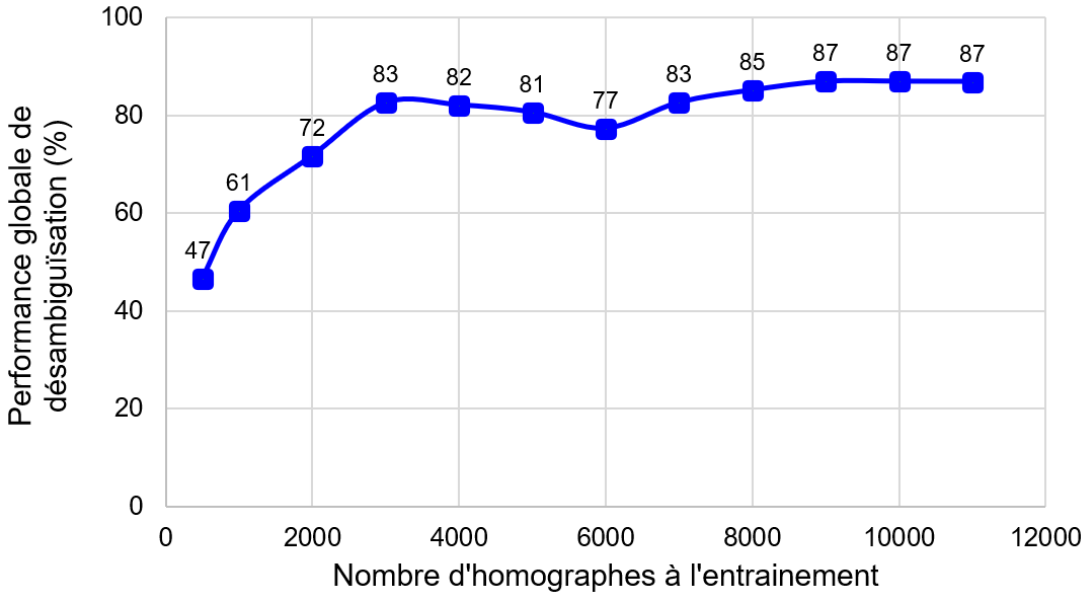


Figure 5.6 : Performance de la désambiguïsation en fonction du nombre d'homographes utilisés lors de l'entrainement manuel

5.3.2.2. Sélection des ensembles d'entrainement et de test

À la section précédente portant sur l'effet du nombre d'homographes utilisés à l'entrainement, les ensembles d'entrainement et les ensembles de tests se chevauchaient. Cette approche n'est généralement pas préconisée, car elle a le potentiel de surestimer la performance obtenue. Dans le cas d'un chevauchement complet, on peut mentionner l'analogie où un étudiant se verrait offrir lors d'un examen exactement les mêmes problèmes qu'on lui avait fournis pour ses devoirs. Dans la section présente, on analyse justement l'effet de la sélection des ensembles d'entrainement et de test sur la performance globale.

Dans un premier lieu, on compare trois approches :

- Chevauchement total : L'ensemble d'entrainement et l'ensemble de test sont les mêmes
  - Cette approche, *non recommandée*, est celle qui a été utilisée plus haut à la Section 5.3.2.1
- Partition du corpus : 70% du corpus sert à l'entrainement, le 30% restant sert au test
  - Ces proportions sont arbitraires, mais ces valeurs (70% vs. 30%) sont souvent utilisées par défaut en apprentissage machine
- Méthode « *k-fold* » : Répétitions du processus avec différentes sections du corpus servant à l'entrainement et au test, suivi d'un calcul de moyenne
  - Ici, une valeur de k=10 est utilisée

Ces trois approches ont été décrites en détail à la Section 4.6.4.1, et illustrées aux Figures 4.25 et 4.26. Aucune explication supplémentaire n'est donc fournie ici.

Pour illustrer l'effet de ces trois approches, le roman « Le Rouge et le Noir » est utilisé comme corpus de référence. Les Figures 5.7, 5.8 et 5.9 fournissent les matrices de confusion obtenues dans les trois cas.

		Classes grammaticales prédites par l'algorithme									
		Verbe	Adj	Nom	Adv	Dét	Pron	Prép	Conj	Inter	Rappel
Classes grammaticales réelles	Verbe	992	92	42							88.1%
	Adj	155	979	51	24	2	2				80.7%
	Nom	59	37	1468	1		1	3	1		93.5%
	Adv	61	18	14	681	6	4		18		84.9%
	Dét		8	21	12	2527	133	11			93.2%
	Pron		10	5	2	18	563		216		69.2%
	Prép	2		6		341		1845			84.1%
	Conj			1	9		62		515		87.7%
	Inter									5	100.0%
	Précision	78.2%	85.6%	91.3%	93.4%	87.3%	73.6%	99.2%	68.7%	100.0%	**86.9%**

Figure 5.7 : Matrice de confusion pour le cas du chevauchement total (l'ensemble d'entrainement est le même que l'ensemble de test)

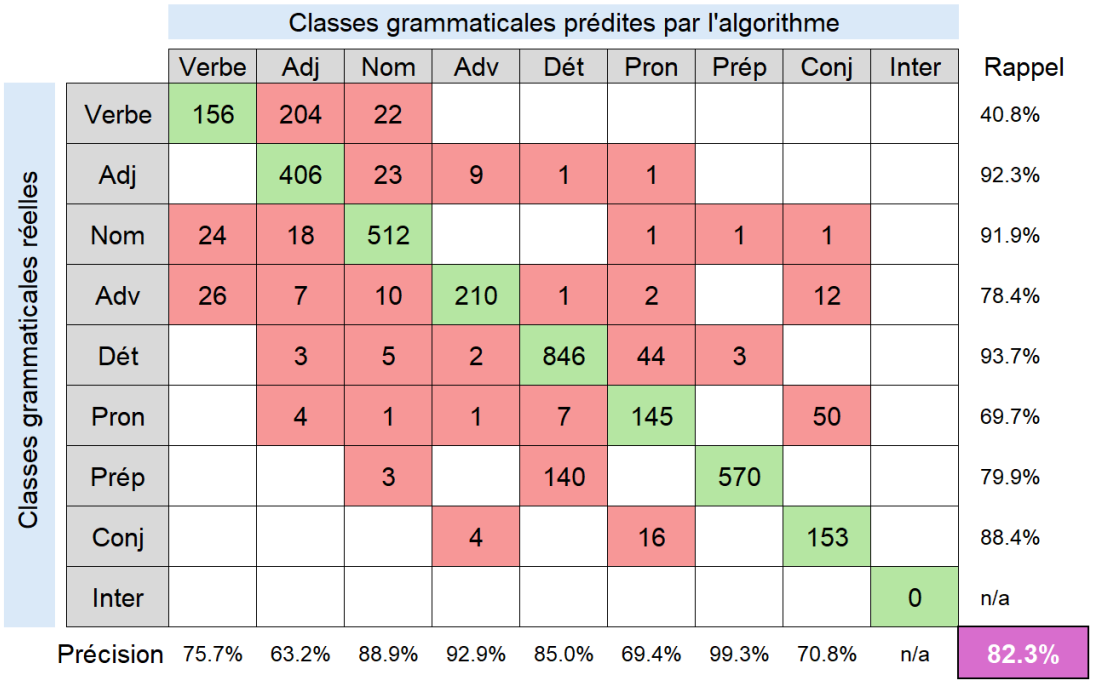


Figure 5.8 : Matrice de confusion pour le cas où 70% des données ont servi à l'entrainement, et le 30% restant à l'évaluation

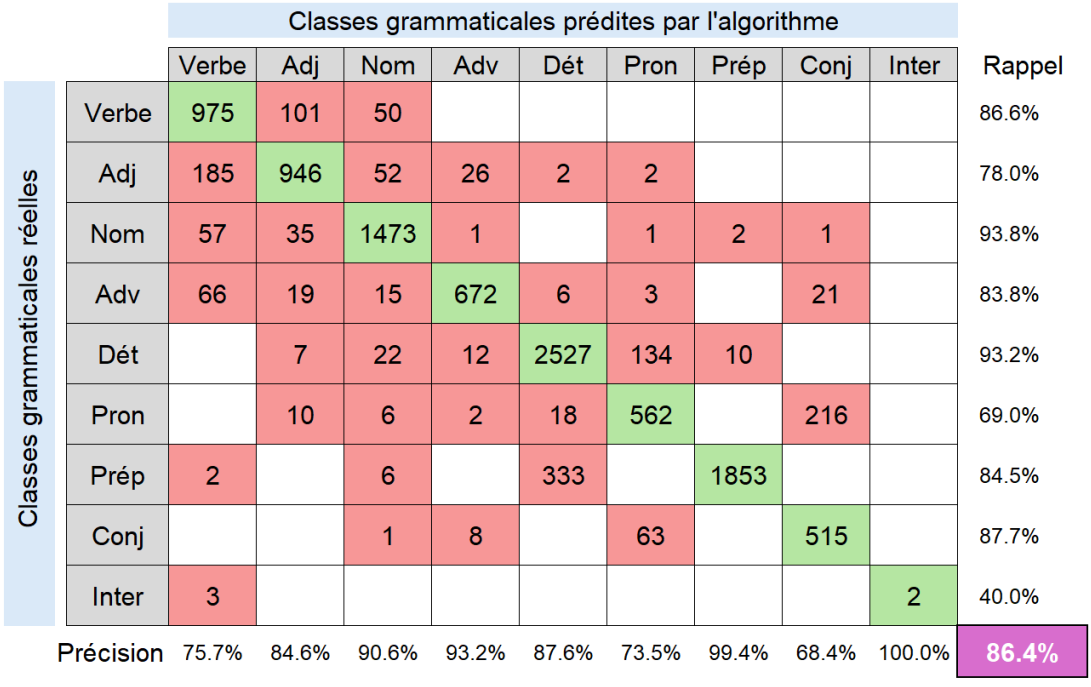


Figure 5.9 : Matrice de confusion pour l'approche *k-fold*, où *k*=10

Comme prévu, la Figure 5.7, correspondant au cas où les deux ensembles se chevauchent, donne une performance globale tout juste sous 87%, en accord avec les résultats présentés à la section précédente où on étudiait l'effet de la taille de l'ensemble d'entrainement. En contraste, la performance globale illustrée à la Figure 5.8 correspondant à la partition « 70-30 » donne une performance bien inférieure, à seulement 82.3%. Cette plus faible performance s'explique par le fait qu'un nombre moindre d'homographes ait servi à l'entrainement (environ 7700). À la Figure 5.6, on avait vu que sous 8000 homographes, on n'avait pas encore atteint le plateau de performance. On voit donc ici que l'approche par partition « 70-30 » n'est pas idéale. Il aurait fallu désambiguïser davantage d'homographes manuellement pour que cette approche soit valable, afin de s'assurer d'atteindre le plateau de la Figure 5.6 avec l'ensemble d'entrainement réduit. On constate aussi que pour le cas avec partition « 70-30 », la matrice de confusion contient beaucoup moins d'entrées, soit environ 30% de ce que contient la matrice de confusion du cas avec chevauchement complet, tel qu'attendu.

Finalement, la Figure 5.9 donne le résultat final obtenu avec la méthode *k-fold*, avec une valeur de *k*=10. On constate que la performance globale est à peine plus faible que celle obtenue avec le chevauchement total (86.4% vs 86.9%). Ce faible écart est rassurant, quand on considère que le cas avec chevauchement total pouvait être interprété comme un maximum théorique. La valeur obtenue avec l'approche *k-fold* s'en approche énormément. Pour le reste de ce mémoire, c'est donc l'approche *k-fold* qui sera adoptée. D'autant plus que celle-ci permet de tester chacun des homographes désambiguïsés manuellement, ce qui offrira une plus grande signification statistique aux analyses subséquentes.

Mais tout comme on l'avait fait pour l'entrainement automatique, il est aussi possible dans le cas manuel de considérer deux corpus différents pour les deux ensembles. En effet, on peut choisir un premier texte pour l'entrainement, et un tout autre texte pour l'évaluation. Dans un tel cas, comme les deux textes ne comportent pas du tout les mêmes données, il n'est pas nécessaire d'effectuer une partition ou d'appliquer l'approche *k-fold*. Le Tableau 5.12 fournit les résultats de performance globale, selon les quatre combinaisons possibles de corpus utilisés pour l'entrainement et pour l'évaluation. Il est à noter que lorsque le même texte est utilisé pour les deux ensembles (cases bleues), l'approche *k-fold* (*k*=10) est appliquée.

On retrouve en premier lieu la valeur de 86.4% déjà discutée plus haut quand le roman « Le Rouge et le Noir » sert à la fois d'ensemble d'entrainement et de test. On constate ensuite que si ce roman est analysé sur la base d'un entrainement effectué avec le roman de science-fiction, une performance légèrement plus faible est obtenue (85.7%). Si on applique la méthode *k-fold* sur le corpus du roman de science-fiction, on obtient un résultat de 87.2%, légèrement supérieur au cas du roman « Le Rouge et le Noir », mais du même ordre de grandeur. Mais si on désambiguïse le roman de science-fiction en utilisant le roman « Le Rouge et le Noir » pour l'entrainement, on obtient une performance légèrement inférieure, soit 87.0%. À la lumière du Tableau 5.12, on en conclut donc, sans surprise, que les meilleurs résultats s'obtiennent quand le même texte est utilisé pour les deux ensembles (avec l'approche *k-fold*). Tout de même, des résultats très semblables sont obtenus quand on utilise deux textes différents pour les ensembles d'entrainement et de test, ce qui est rassurant. Cela semble impliquer que tout corpus utilisé pour l'entrainement, dans la mesure où il est composé de suffisamment d'homographes désambiguïsés manuellement, peut se montrer efficace pour désambiguïser tout autre texte. Il est aussi intéressant de mentionner, une fois de plus, que les performances obtenues avec l'entrainement manuel (Tableau 5.12) sont supérieures aux performances obtenues précédemment avec l'entrainement automatique (Tableau 5.11). Cette observation n'est pas surprenante, considérant l'information fiable et additionnelle fournir par l'humain à l'algorithme de désambiguïsation dans le cas de l'apprentissage manuel.

Tableau 5.12 : Performance globale de la désambiguïsation avec l'approche manuelle, en considérant les 4 combinaisons possibles d'ensembles d'entrainement et de test. Dans les cas où le même texte est utilisé pour les deux ensembles (cases bleues), l'approche *k-fold* est utilisée

		Corpus utilisé pour l'évaluation	
		Le Rouge et le Noir	Science-Fiction
Corpus utilisé pour l'entrainement	Le Rouge et le Noir	86.4%	87.0%
	Science-Fiction	85.7%	87.2%

5.3.2.3. Performance des tests spécialisés

L'algorithme de désambiguïsation de base appliqué jusqu'à présent, est basé sur les caractéristiques générales listées à l'Annexe B. Cet algorithme cherche à déterminer à quelle classe grammaticale (parmi les neuf classes considérées) appartient chaque homographe. Cette approche se base sur le fait qu'à chaque classe grammaticale correspond une « signature » distincte en ce qui concerne les caractéristiques globales, et que tout mot appartenant à une classe pourrait ainsi en substituer un autre de la même classe dans le texte. Cette approche s'est avérée assez efficace, si on en juge par les niveaux de performance respectables de la désambiguïsation illustrés aux Tableaux 5.11 (cas automatique) et 5.12 (cas manuel).

Mais certains homographes en particulier n'ont pas été désambiguïsés aussi efficacement que ce que les moyennes de ces deux tableaux indiquent. Tel que décrit à la Section 4.6.5, on a donc, pour certains de ces mots, développé des caractéristiques distinctes ne s'appliquant qu'à un mot en particulier. C'est donc dire qu'on a appliqué des équations de régression logistique ne s'appliquant qu'à un seul mot. Les homographes causant problème, identifiés dans ce projet pour le roman « Le Rouge et le Noir », sont identifiés au Tableau 5.13. Tous ces mots sont suffisamment nombreux dans le corpus pour qu'une analyse de régression logistique « spécialisée » puisse être efficace. Pour ce qui est des homographes « si », « s' » et « de/d' », les performances de désambiguïsation n'étaient pas mauvaises du tout (en particulier pour « s' ») en utilisant l'algorithme de base. Mais des caractéristiques ont facilement pu être identifiées pour améliorer davantage la performance, ce que l'on verra un peu plus loin.

Tableau 5.13 : Homographes pour lesquels des tests spécialisés ont été mis au point pour ce projet

Homographe	Classes impliquées	Fréquence dans le roman « Le Rouge et le Noir »	Performance de désambiguïsation
tout	Adjectif, nom, adverbe, déterminant ou pronom	134	50.7%
que / qu'	Pronom ou conjonction	791	66.2%
même	Adjectif ou adverbe	80	65.0%
autre / autres	Adjectif ou pronom	37	67.6%
si	Adverbe ou conjonction	145	80.0%
de / d'	Déterminant ou préposition	2154	84.1%
s'	Pronom ou conjonction	172	93.0%

Pour chacun de ces homographes, des caractéristiques spécifiques ont été mises au point, se rapportant précisément à l'usage de ces mots dans les textes en français. Plusieurs de ces caractéristiques ont été inspirées des définitions très complètes de ces mots fournies dans le dictionnaire en ligne québécois Usito (2024). En effet, pour chacun de ces mots, Usito liste une foule de cas d'utilisation du mot, fournissant ainsi des indices sur les meilleures caractéristiques à employer. À l'Annexe C, on présente en exemple un extrait de l'entrée d'Usito pour le mot « que ». Finalement, à l'Annexe D, on fournit la liste de toutes les caractéristiques qui ont été utilisées pour les homographes apparaissant au Tableau 5.13.

Au moment de l'entrainement, un ensemble distinct est donc créé pour chacun de ces homographes, à partir duquel des équations de régression logistique ont été créées. Au moment de l'évaluation, si un homographe fait partie de la liste des mots du Tableau 5.13, c'est à partir de ces régressions logistiques spécialisées que la classe grammaticale sera déterminée, plutôt que sur la base des régressions générales obtenues sur la base de tous les mots du corpus. Il va de soi que les tests spécialisés ne s'appliquent qu'au cas d'entrainement manuel, puisque tous les mots du Tableau 5.13 sont des homographes, et que l'entrainement automatique s'effectue uniquement sur la base des *non* homographes.

Le Tableau 5.14 fournit les performances de désambiguïsation obtenues avec et sans l'apport de ces tests spécialisés, en utilisant le roman « Le Rouge et le Noir ». On constate que dans tous les cas, les tests spécialisés ont permis d'augmenter la performance de désambiguïsation. Dans certains cas l'amélioration est marquée, de l'ordre de 30% de différence, pour les mots « tout », « même » et « autre ». On remarque aussi qu'on a obtenu un score parfait pour les homographes « autre », « si » et « s' » grâce à ces tests spécialisés.

En comparant le total des mots réussis sans et avec les tests spécialisés, en considérant tous les homographes du Tableau 5.14, on constate que les tests spécialisés ont permis de correctement identifier 524 mots supplémentaires (3280 – 2756 = 524). En se concentrant sur uniquement les mots du Tableau 5.14, on obtient une performance de désambiguïsation de 93.4% avec les tests spécialisés, comparativement à une bien plus faible valeur de 78.5% sans ces tests. La performance de 93.4% pour ces homographes soumis à des tests spécialisés est aussi bien supérieure à la performance globale de désambiguïsation pour le corpus en entier, qui s'élevait à 86.4%. On peut donc en conclure que les tests spécialisés se sont avérés très utiles.

Tableau 5.14 : Performance de désambiguïsation des homographes pour lesquels des tests spécialisés ont été mis au point – comparaison sans et avec les tests spécialisés

	Sans test spécialisé			**Avec tests spécialisés**		
	Réussis	Total	Performance	Réussis	Total	Performance
tout	68	134	50.7%	111	134	82.8%
que/qu'	524	791	66.2%	664	791	83.9%
même	52	80	65.0%	79	80	98.8%
autre/autres	25	37	67.6%	37	37	100.0%
si	116	145	80.0%	145	145	100.0%
de/d'	1811	2154	84.1%	2072	2154	96.2%
s'	160	172	93.0%	172	172	100.0%
Combiné	2756	3513	78.5%	3280	3513	93.4%

Mais la performance des tests spécialisés n'est pas parfaite pour autant. Les tableaux de la Figure 5.10, baptisés ici « matrices de *contribution* », fournissent des détails concernant chacun des mots du Tableau 5.14. Au sein de ces matrices de contribution, on classifie chaque occurrence du mot dans le corpus, selon qu'il a été bien ou mal classifié avec et sans l'utilisation des tests spécialisés. Les cases grises correspondent aux cas où la classification n'a pas été affectée par les tests spécialisés. Ou bien le mot est mal classifié dans les deux cas, ou bien il a été bien classifié dans les deux cas. Ces scénarios (cases grises) ne nous intéressent pas ici. En revanche, les cases vertes correspondent aux cas où les homographes ont été mal classifiés *sans* les tests spécialisés, mais bien classifiés *avec* les tests spécialisés. Idéalement, ce chiffre devrait être le plus élevé possible. Malheureusement, dans certains scénarios, la prédiction avec tests spécialisés s'est avérée fausse, alors que la prédiction sans les tests spécialisés était correcte. Ces scénarios illustrent des failles dans la performance des tests spécialisés, et il est évidemment souhaitable que ces cas (cases rouges) soient le moins élevés possible. On espère donc une bien plus grande valeur pour les cases vertes, en comparaison avec les cases rouges, ce qu'on observe d'ailleurs. Cette comparaison entre les valeurs des cases vertes et rouges permet donc de quantifier la contribution des tests spécialisés à l'amélioration de la performance de désambiguïsation.

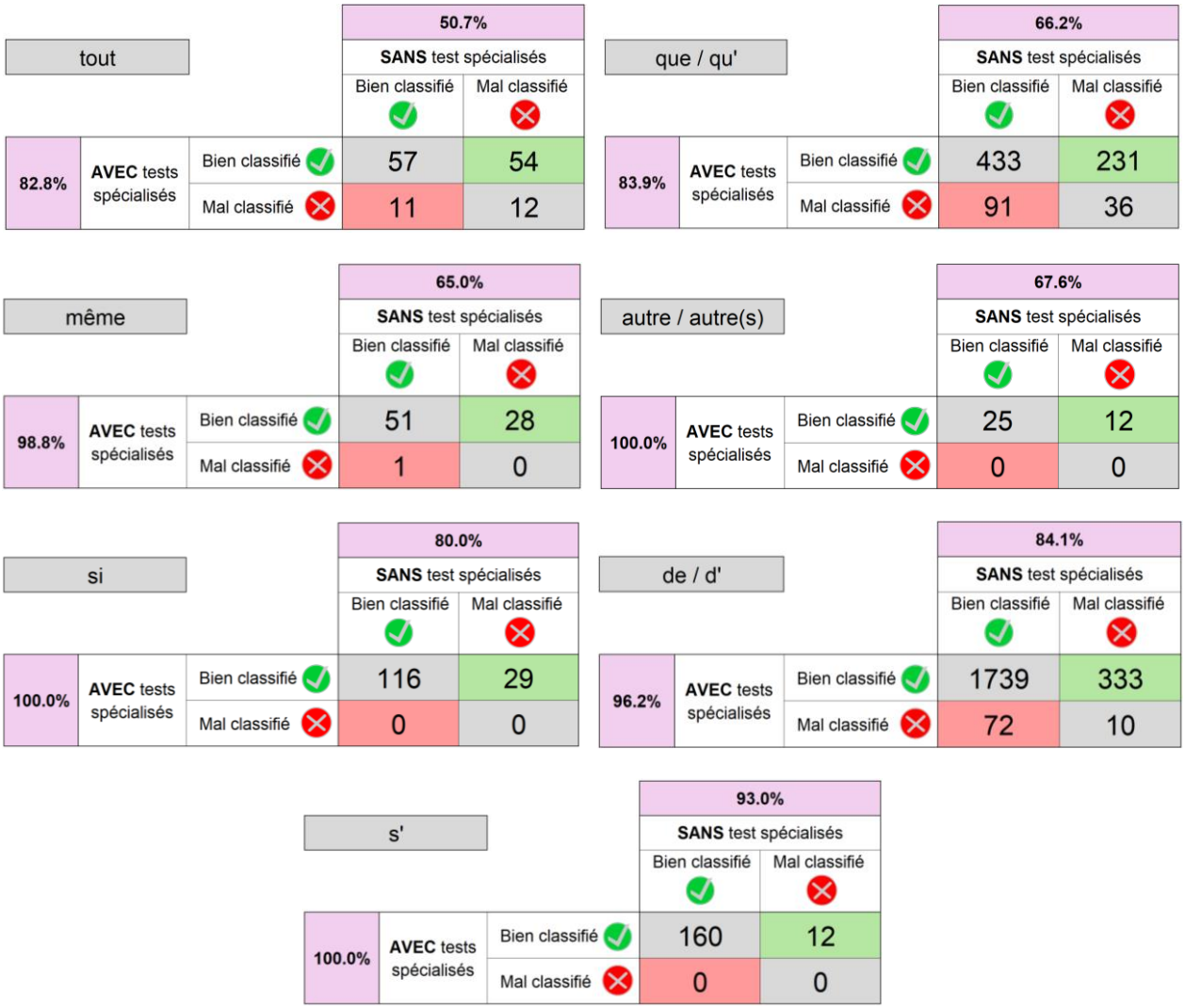


Figure 5.10 : Matrices de contribution pour tous les homographes du Tableau 5.14, illustrant la performance de désambiguïsation avec et sans les tests spécialisés

Pour les homographes « autre/autres(s) », « si » et « s' », on n'observe aucun scénario où les tests spécialisés ont mal classifié un mot qui était auparavant bien classifié. Pour l'homographe « même », on n'observe qu'un seul tel cas. En revanche, pour les homographes « tout », « que/qu' » et « de/d' », le nombre de scénarios dans les cases rouges est plus élevé. Mais malgré tout, même dans ces trois cas, les valeurs dans les cases rouges sont beaucoup plus grandes, par des facteurs de l'ordre de 2.5 à 5. C'est donc dire que même pour ces homographes, la contribution des tests spécialisés a été démontrée. La Figure 5.11 résume l'effet combiné des tests spécialisés sur tous les mots illustrés au Tableau 5.14, une fois de plus sous la forme d'une matrice de contribution.

Jetons maintenant un coup d'œil à la contribution des tests spécialisés sur la performance de la désambiguïsation sur le corpus *en entier*. Sans test spécialisé, l'algorithme général avait correctement classifié 9526 homographes sur un total de 11023, pour une performance de 86.4%. En rajoutant les 524 homographes bien identifiés grâce aux tests spécialisés, on pourrait s'attendre à une performance de 91.7% sur la base de cette équation :

$$\frac{9526 + 524}{11\,023} = 0.917 = 91.7\%$$

Mais dans les faits, la performance globale pour le corpus entier obtenue est supérieure, se situant plutôt à 92.9%. Nous sommes donc ici témoins d'un effet de levier, car la correction de 524 homographes mal classifiés parmi la liste du Tableau 5.13 a résulté en la correction de 194 *autres* homographes qui étaient auparavant eux aussi mal classifiés. Autrement dit, pour chaque homographe corrigé *directement* par un test spécialisé, on se retrouve à corriger en moyenne 0.37 homographe additionnel. Ce résultat s'explique aisément, car en corrigeant la classification d'un homographe dans une phrase donnée, on permet par la suite un meilleur calcul des caractéristiques des autres homographes inclus dans cette même phrase. En retour, de meilleures classifications mènent à d'autres meilleures classifications pour les mots adjacents. On peut illustrer cet effet en étudiant la phrase suivante :

« J'ai hâte de manger »

Si l'algorithme de base (sans les tests spécialisés) classifiait erronément l'homographe « de » dans cette phrase comme étant un déterminant plutôt qu'une préposition, les caractéristiques pour l'homographe suivant (« manger ») feraient en sorte que celui-ci aurait davantage de chances d'être classifié comme un nom commun. En effet, à la suite d'un déterminant, on retrouve bien plus souvent un nom commun qu'une forme verbale. Cependant, si on activait le test spécialisé et que celui-ci contredisait l'algorithme de base, statuant correctement que l'homographe « de » est une préposition dans ce contexte, il deviendrait beaucoup plus probable que l'homographe « manger » soit cette fois classifié (correctement) comme une forme verbale. En effet, à la suite d'une préposition, on retrouve souvent une forme verbale à l'infinitif.

Tests spécialisés			78.5%	
			SANS test spécialisés	
			Bien classifié ✓	Mal classifié ✗
93.4%	**AVEC** tests spécialisés	Bien classifié ✓	2581	699
		Mal classifié ✗	175	58

Figure 5.11 : Matrice de contribution pour tous les homographes du Tableau 5.14, illustrant la performance de désambiguïsation avec et sans les tests spécialisés

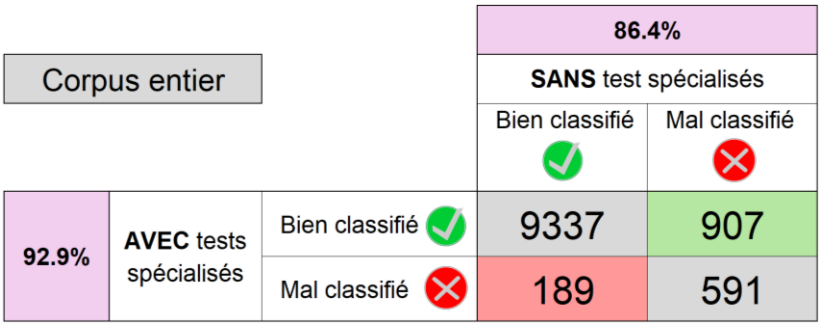


Figure 5.12 : Matrice de contribution pour le corpus en entier (roman « Le Rouge et le Noir »), illustrant la performance de désambiguïsation avec et sans les tests spécialisés

On clôt finalement la présente section en présentant la matrice de contribution globale des tests spécialisés (Figure 5.12), en observant l'effet total (direct et indirect) de ces tests. On constate une fois de plus que la performance des tests spécialisés n'est pas parfaite, car on retrouve 189 homographes (cases rouges) qui étaient auparavant bien classifiés et qui se sont retrouvés mal classifiés après l'application des tests spécialisés. Mais ces 189 erreurs de classification sont amplement compensées par les 907 homographes qui au contraire, se sont retrouvés à être bien classifiés grâce aux tests spécialisés (directement ou indirectement). Grâce à l'ajout des tests spécialisés, la performance globale de désambiguïsation du roman « Le Rouge et le Noir » est donc passée de 86.4% à une bien plus grande valeur de 92.9%. Il est à noter que ces valeurs sont basées sur l'évaluation par l'approche *k-fold* où *k*=10.

La même analyse de tests spécialisés a été effectuée sur le roman de science-fiction. Le détail des résultats n'est pas présenté ici. Mais pour ce corpus aussi, les tests spécialisés se sont avérés très utiles, puisque la performance globale pour le corpus en entier est passée de 87.2% à 91.9%. Ces valeurs (sans et avec les tests spécialisés) sont très comparables entre les deux corpus.

5.3.2.4. Performance du test de participes passés

Dans le cadre de ce projet, tous les participes passés sont considérés comme des homographes, tel que discuté à la Section 4.1.4. En effet, ceux-ci sont considérés soit comme des formes verbales, soit comme adjectifs, lorsqu'ils sont employés seuls. Donc, sans exception, tout participe passé se retrouvant dans un texte se doit d'être désambiguïsé. Afin de faciliter cette désambiguïsation, les participes passés ont été regroupés selon le nombre de classes grammaticales et fonctions possibles pour chacun. On les a ainsi regroupés sous quatre « scénarios », tel que discuté à la Section 4.7.1. Les Tableaux 5.15 et 5.16 illustrent en premier lieu le nombre d'occurrences pour chacun de ces quatre scénarios, quelques exemples pour chaque cas, ainsi que la performance de base de désambiguïsation, pour les deux corpus de référence étudiés ici. Les performances relativement basses listées à la dernière colonne de ces deux tableaux démontrent la pertinence de mettre au point un algorithme ayant pour but d'améliorer la classification des participes passés. Le fonctionnement de cet algorithme a été illustré à la Figure 4.36, et ne sera pas décrit davantage ici, autrement que pour dire qu'il fait appel à un test analytique permettant, dans certains cas, d'éliminer la possibilité qu'un participe passé soit considéré comme un adjectif (participe passé employé seul).

Tableau 5.15 : Occurrences des participes passés et performance de désambiguïsation selon les quatre scénarios, pour le roman « Le Rouge et le Noir » (portion lemmatisée manuellement)

Scénario	Description	Nombre total	Nombre distinct	Exemples	Performance de désambiguïsation
1	Participe passé seulement	785 (52%)	435 (73%)	bâties, ruinées, abrité	74.4%
2	Participe passé ou verbe conjugué	199 (13%)	34 (5.7%)	dites, remplis, écrit	84.4%
3	Participe passé ou autre classe grammaticale	181 (12%)	21 (3.5%)	fait, point, mépris, nuit, trait	87.3%
4	Participe passé, verbe conjugué ou autre classe grammaticale	353 (23%)	108 (18%)	partie, pavé, vue, faits, parties, rangées	88.4%

Tableau 5.16 : Occurrences des participes passés et performance de désambiguïsation selon les quatre scénarios, pour le roman de science-fiction (portion lemmatisée manuellement)

Scénario	Description	Nombre total	Nombre distinct	Exemples	Performance de désambiguïsation
1	Participe passé seulement	684 (56%)	436 (75%)	codée, rassuré, resté, réussi	74.7%
2	Participe passé ou verbe conjugué	100 (8.1%)	36 (6.2%)	atteint, plaint, construit, dit	77.0%
3	Participe passé ou autre classe grammaticale	196 (16%)	20 (3.5%)	fait, point, conduit, produit	58.2%
4	Participe passé, verbe conjugué ou autre classe grammaticale	251 (20%)	87 (15%)	pensées, étendue, ralenti	75.7%

Les Tableaux 5.15 et 5.16 illustrent uniquement la performance de désambiguïsation *avant* l'application du test pour les participes passés. Tout comme nous l'avions fait à la section précédente, nous allons maintenant présenter les « matrices de contribution », qui servent à illustrer l'efficacité de l'algorithme pour les participes passés. Ces matrices sont affichées à la Figure 5.13 pour les deux corpus sous étude. Cette figure se limite uniquement aux homographes pouvant être des participes passés. On constate que bien plus de participes passés ont été mieux classés (cases vertes) que moins bien classés (cases rouges), suivant l'application du test de participes passés. Les ratios entre ces deux cas sont de 19.2 et 18.8 pour les deux corpus, respectivement. Mais surtout, on note une nette amélioration de la performance de désambiguïsation, soit plus de 14% pour le roman « Le Rouge et le Noir », et plus de 17% pour le roman de science-fiction.

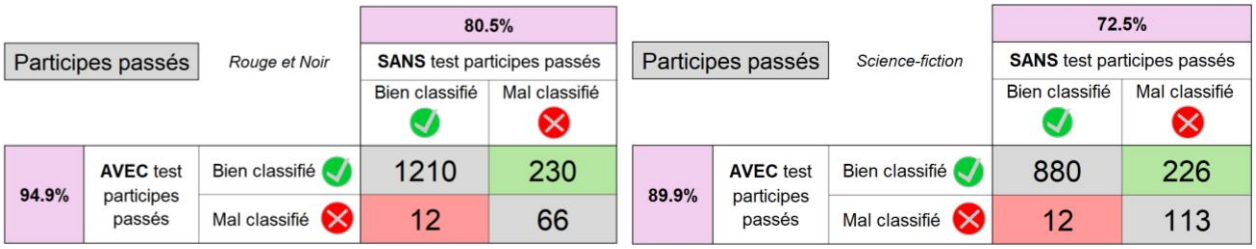


Figure 5.13 : Matrices de contribution pour *tous les homographes pouvant être des participes passés*, illustrant la performance de désambiguïsation avec et sans le test de participes passés, pour le roman « Le Rouge et le Noir » (à gauche) et le roman de science-fiction (à droite)

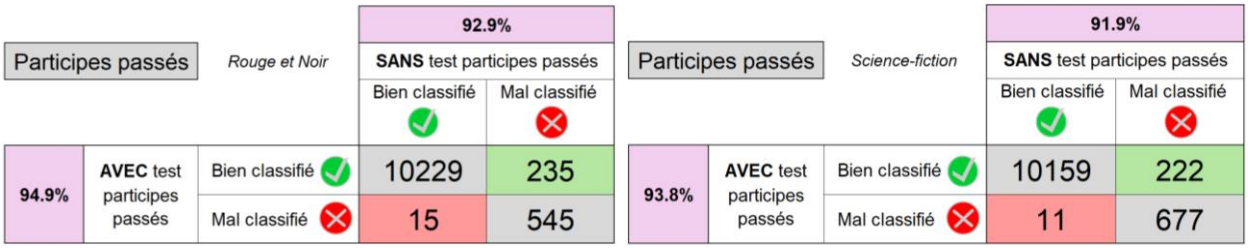


Figure 5.14 : Matrices de contribution pour *tous les homographes du corpus*, illustrant la performance de désambiguïsation avec et sans le test de participes passés, pour le roman « Le Rouge et le Noir » (à gauche) et le roman de science-fiction (à droite)

Les matrices de contribution de la Figure 5.14 concernent aussi le test de participes passés, mais considérant cette fois *tous les mots du corpus*. En comparant ces figures avec les précédentes (Figure 5.13), on n'observe qu'un bien faible effet de levier. En effet, pour le roman « Le Rouge et le Noir », tandis qu'une amélioration nette de 218 homographes bien classés est observée en ne considérant que les homographes pouvant être des participes passés, ce chiffre ne monte qu'à 220 en considérant tous les homographes du corpus. Dans le cas du roman de science-fiction, l'amélioration nette de 214 homographes bien classés se produit au contraire par une légère diminution, à 211, en considérant le corpus entier.

Toujours est-il, que malgré cette absence d'effet de levier, qu'on avait observé à la Section 5.3.2.3 pour les tests spécialisés, la contribution du test de participes passés s'est avérée bénéfique pour les deux corpus. La Figure 5.15 confirme d'ailleurs que pour les deux corpus et pour les quatre scénarios, la performance de désambiguïsation s'est améliorée en appliquant le test de participes passés (barres vertes) en comparaison aux cas sans ce test (barres bleues). C'est au niveau du Scénario 1 que les améliorations sont les plus marquées. On se rappellera que ce scénario n'implique aucun apprentissage machine. Les plus faibles résultats, autant avec que sans le test de participes passés, ont été obtenus pour le Scénario 3, pour le roman de science-fiction (autour de 60%). Ce faible résultat est dû au faible taux de réussite pour l'homographe « point » classifié sous le Scénario 3, utilisé abondamment sous sa forme « nominale » dans ce roman. En effet, en ne considérant que la partie lemmatisée du corpus, seules 14 occurrences du mot « point » ont été correctement désambiguïsées, sur les 73 occurrences répertoriées.

Pour les deux corpus, l'amélioration globale sur le corpus entier peut sembler minime. Mais on se rapproche de plus en plus d'une valeur de 100%, et ces derniers points de pourcentage sont les plus difficiles à gruger.

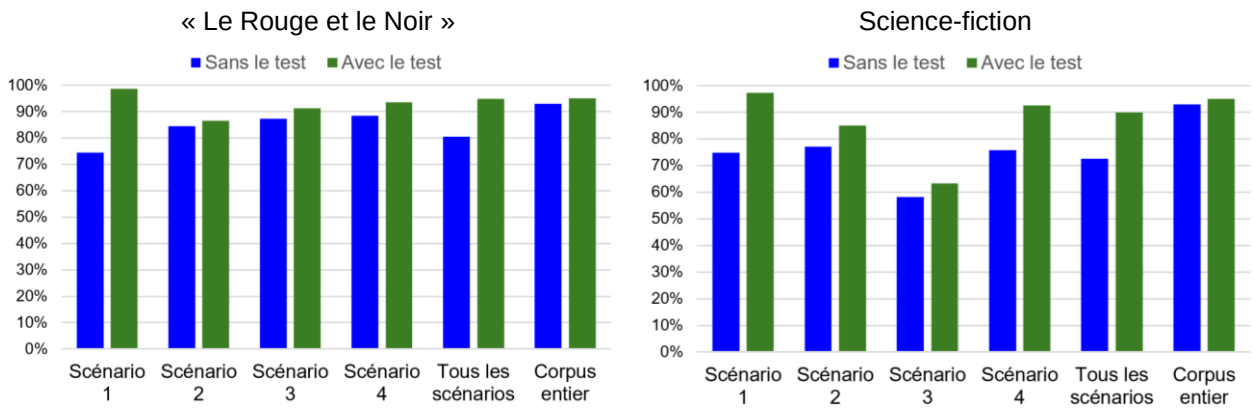


Figure 5.15 : Performances de désambiguïsation des homographes pouvant être des participes passés, pour le roman « Le Rouge et le Noir » (à gauche) et pour le roman de science-fiction (droite). Les quatre scénarios distincts sont représentés, suivis de la performance moyenne pour tous les scénarios. Finalement, l'effet global du test de participes passés est illustré (colonne la plus à droite), en considérant tous les homographes du corpus

5.3.2.5. Performance des tests de locutions

Tout le comme le test de participes passés, qui n'y fait pas directement appel, le test de locutions n'est pas non plus basé sur l'apprentissage machine. Tel que décrit à la Section 4.7.2, l'algorithme pour les locutions recherche des combinaisons particulières de lemmes qu'on retrouve fréquemment dans la langue, qu'on appelle « locutions » ou « syntagmes ». En présence de telles combinaisons, dont l'un des deux mots est un homographe, l'algorithme de désambiguïsation lui assigne alors la classe grammaticale appropriée, de façon déterministe. La Section 4.7.2 incluait des exemples tels que « faire place », « rendre grâce », ou le deuxième mot, un homographe, doit être classifié comme un nom. Mais l'algorithme inclut aussi certains autres cas tels que :

- Les homographes « avoir » et « être » suivis d'un participe passé : ces mots doivent être classifiés comme verbes, et non comme des noms
- L'homographe « fait » suivi d'un infinitif (« fait mouvoir ») : ce mot doit être classifié comme un verbe et non comme un nom
- L'homographe « point » : précédé d'un déterminant, doit être classifié comme un nom, mais doit être classifié comme un adverbe dans des expressions telles que « n'est-ce point »
- L'homographe « pas » : doit être classifié comme un adverbe, dans des expressions telles que « n'est-ce pas » ou « ne pas », plutôt que comme un nom
- Le mot « personne » : doit être classifié comme un pronom lorsque précédé de « à », « par », « de » et « que », plutôt que comme un nom
- Les homographes « que » et « qu' » : doivent être classifiés comme pronoms, suivant le mot « ce », plutôt que comme déterminants
- L'homographe « comment » : doit être classifié comme un nom, lorsque précédé du déterminant « le », plutôt que comme adverbe
- Les homographes « saint » et « sainte » : doivent être classifiés comme noms propres, lorsque précédés de noms propres et autres cas précis (Saint-Louis, Saint-Sauveur), plutôt que comme adjectifs
- L'homographe « bouche » : doit être classifié comme un nom, dans la locution « bouche bée »

La langue française regorge de tels exemples de locutions et syntagmes impliquant des homographes, mais seul un nombre limité a été considéré ici. La performance de ce test est illustrée en premier lieu à la Figure 5.16, à l'aide d'une matrice de contribution bâtie pour chacun des deux corpus utilisés dans cette étude. On y constate que ces tests se sont avérés efficaces. En effet, ils ont permis de correctement classifier certains homographes qui ne l'étaient pas précédemment (cases vertes), sans jamais incorrectement classifier des homographes qui étaient déjà bien classifiés (les cases rouges sont égales à zéro). Avant l'application du test de locution, l'efficacité de la désambiguïsation de ces homographes faisant potentiellement partie d'une locution n'était que de 66.4% pour le roman « Le Rouge et le Noir », et de 45.6% pour le roman de science-fiction.

Le Tableau 5.17 fournit la liste détaillée des 37 homographes du roman « Le Rouge et le Noir » et des 92 homographes du roman de science-fiction ayant été correctement désambiguïsés grâce au test de locutions. Ces homographes avaient auparavant été incorrectement désambiguïsés à l'aide des autres algorithmes (apprentissage machine). Il est à noter que dans le cas du roman de science-fiction, l'homographe « point » a été corrigé pas moins de 58 fois. Le test de locutions a donc permis de corriger le problème avec l'homographe « point » qui avait été mentionné précédemment (Section 5.3.2.4) dans le contexte de la désambiguïsation des participes passés (« point » est le participe passé du verbe « poindre »).

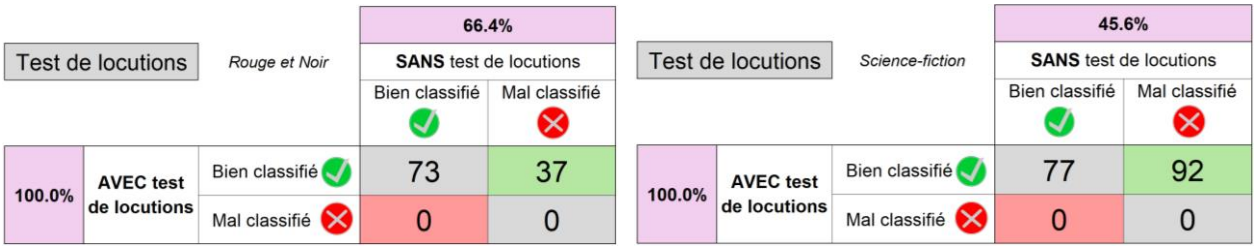


Figure 5.16 : Matrices de contribution pour *tous les homographes pouvant faire partie d'une locution*, illustrant la performance de désambiguïsation avec et sans le test de locutions, pour le roman « Le Rouge et le Noir » (à gauche) et le roman de science-fiction (à droite)

Tableau 5.17 : Occurrences des mots faisant partie de locutions pour les deux corpus de référence (portions lemmatisées manuellement), correctement désambiguïsées grâce au test de locutions

Le Rouge et le Noir (37 cas)

Locution	#	Locution	#
fait	5	garde	2
avoir	4	mine	1
envie	3	semblant	1
pas	3	part	1
saint	3	comment	1
place	3	grâce	1
être	3	bataille	1
compte	2	court	1
personne	2		

Science-fiction (92 cas)

Locution	#	Locution	#
point	58	pas	2
comment	6	partie	1
compte	6	bouche	1
fait	5	semblant	1
avoir	3	que	1
place	3	visite	1
signe	3	mine	1

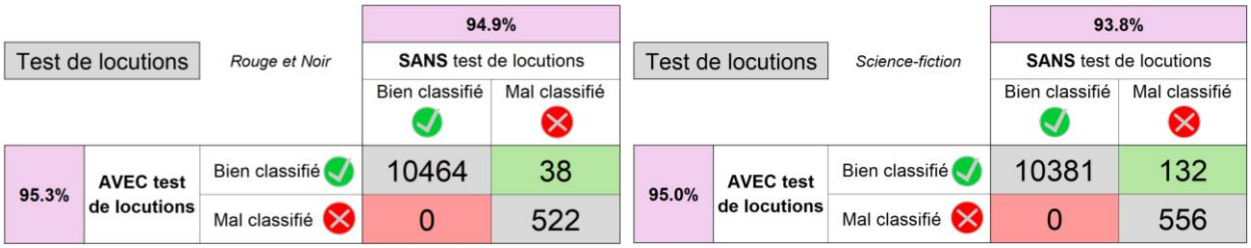


Figure 5.17 : Matrices de contribution pour *tous les homographes du corpus*, illustrant la performance de désambiguïsation avec et sans le test de locutions, pour le roman « Le Rouge et le Noir » (à gauche) et le roman de science-fiction (à droite)

Alors que la Figure 5.16 se concentre uniquement sur les mots du corpus pouvant potentiellement faire partie d'une locution, la Figure 5.17 considère quant à elle l'effet du test de locutions sur la performance globale de la désambiguïsation de tout le corpus. On constate que dans le cas du roman « Le Rouge et le Noir », un seul homographe additionnel a été bien classifié, au-delà de ceux directement affectés par le test (38 corrections, en comparaison avec 37). Mais dans le cas du roman de science-fiction, on constate que la correction des 92 locutions illustrée à la Figure 5.16 et détaillée au Tableau 5.17 a mené à la correction de 132 homographes en tout, une augmentation de 40. C'est donc dire qu'un certain effet de levier a ici été observé. En effet, la correction des 92 homographes obtenue directement par l'application du test a permis de mieux classifier d'autres homographes à proximité dans la phrase.

La Figure 5.17 nous permet aussi de constater que l'application du test de locutions a permis de légèrement augmenter la performance de désambiguïsation globale pour les deux corpus : de 94.9% à 95.3% pour « Le Rouge et le Noir », et de 93.8% à 95.0% pour le roman de science-fiction.

5.3.2.6. Performance des tests statistiques sur les verbes

Le fonctionnement de l'algorithme de statistiques sur les verbes a été décrit en détail à la Section 4.7.3. Cet algorithme identifie d'abord des homographes pouvant être ou bien un verbe conjugué ou un nom commun, et tente ensuite de déterminer les chances que cet homographe soit un verbe. Pour y arriver, l'algorithme compare la fréquence du temps et de la personne du verbe en jeu, avec les temps et les personnes de tous les autres verbes du corpus. La présente section ne fait que fournir les résultats obtenus à l'aide de ce test statistique, appliqué aux deux corpus sous étude. À la Figure 5.18, on illustre les deux matrices de contribution pour ce test, appliqué aux deux corpus. Ces deux matrices se concentrent uniquement sur les homographes de type « verbe vs. nom ». On constate que le test statistique a permis de légèrement augmenter la performance de la désambiguïsation de ce type d'homographes, pour les deux corpus.

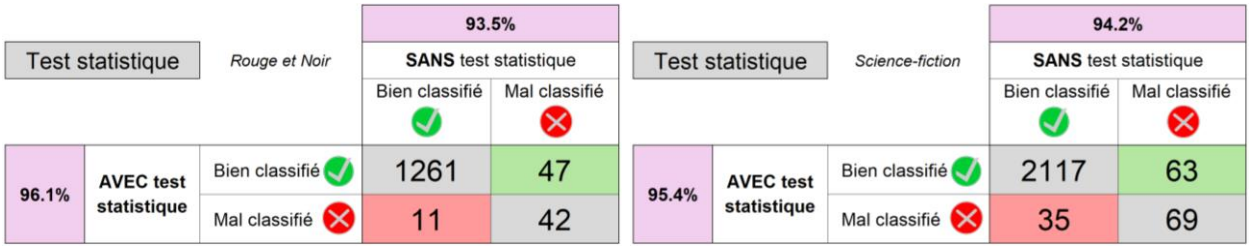


Figure 5.18 : Matrices de contribution pour *tous les homographes pouvant être des verbes mais identifiés comme ne l'étant sans doute pas*, illustrant la performance de désambiguïsation avec et sans le test statistique pour verbes, pour le roman « Le Rouge et le Noir » (à gauche) et le roman de science-fiction (à droite)

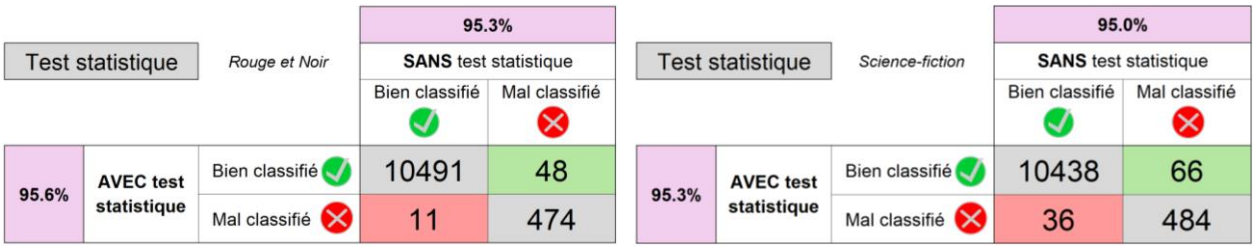


Figure 5.19 : Matrices de contribution pour *tous les homographes du corpus*, illustrant la performance de désambiguïsation avec et sans le test de statistiques pour les verbes, pour le roman « Le Rouge et le Noir » (à gauche) et le roman de science-fiction (à droite)

Toutefois, on constate aussi que le nombre de cas où l'algorithme a incorrectement classifié comme un nom un homographe qui aurait dû être un verbe (cases rouges à la Figure 5.18) n'est pas négligeable. En particulier, pour le roman de science-fiction, on a noté 35 erreurs de classification pour 63 corrections de classification. L'efficacité de ce test statistique est donc limitée, mais son effet global demeure tout de même positif. Il faut se rappeler que ce test n'est basé que sur des hypothèses statistiques et donc aucunement sur la structure grammaticale des phrases où chacun de ces homographes apparait. Il ne faut donc pas se surprendre de son efficacité somme toute limitée, en comparaison avec d'autres tests appliqués dans ce projet.

La Figure 5.19 fournit les matrices de contribution pour le test statistique, cette fois en considérant tous les mots des deux corpus. On remarque dans les deux cas une augmentation de la performance globale de la désambiguïsation d'environ 0.3%. Bien que faible, cette augmentation justifie tout de même de conserver le test statistique au sein de l'algorithme global de désambiguïsation.

À l'Annexe F, on fournit des exemples précis de classifications correctes (cases vertes) et de classifications incorrectes (cases rouges).

5.3.2.7. Performance en fonction du nombre de classes grammaticales

En grande majorité, les homographes de la langue française n'impliquent que deux classes grammaticales possibles. Par exemple, l'homographe « demande » peut être soit un verbe, soit un nom, mais ne peut appartenir à aucune autre classe. Dans de tels cas n'impliquant que deux classes grammaticales, une approche d'apprentissage machine basée sur la régression logistique *binaire* est tout à fait appropriée. Mais certains homographes peuvent appartenir à plus de deux classes grammaticales, comme l'homographe « ferme » qui peut être ou bien un verbe, un adjectif ou un nom. Dans de tels cas, la régression logistique binaire ne peut être appliquée directement. Une approche différente, décrite à la Section 4.6.2.2.2 a donc été mise au point pour désambiguïser les homographes comprenant plus de deux classes grammaticales possibles. On ne mentionnera pas à nouveau ici le détail du fonctionnement de cette approche, se contentant plutôt de comparer l'efficacité de désambiguïsation obtenue en fonction du nombre de classes grammaticales possibles pour chaque homographe.

Si la désambiguïsation s'effectuait tout simplement *au hasard*, on s'attendrait à une performance d'environ 50% pour les cas avec deux classes grammaticales possibles, et d'environ 33% pour les cas avec trois classes possibles, et ainsi de suite pour les cas avec quatre classes possibles (l'homographe « point ») et cinq classes possibles (l'homographe « tout »). C'est donc dire qu'uniquement sur cette base, on peut s'attendre à une performance de désambiguïsation moindre, lorsque le nombre de classes grammaticales possibles est supérieur à deux. C'est ce qu'on tente de démontrer ici. Le Tableau 5.18 fournit justement la performance de désambiguïsation des homographes en fonction du nombre de classes grammaticales possibles, pour les deux corpus.

Tableau 5.18 : Performance de la désambiguïsation en fonction du nombre de classes grammaticales possibles des homographes, pour les deux corpus

	Le Rouge et le Noir		Science-fiction	
Nombre de classes grammaticales	Nombre d'occurrences	Performance de la désambiguïsation	Nombre d'occurrences	Performance de la désambiguïsation
2	10 207 (93%)	96.0%	10 326 (93%)	95.8%
3	632 (5.7%)	92.1%	516 (4.7%)	88.2%
4 (« point »)	51 (0.46%)	98.0%	73 (0.66%)	98.6%
5 (« tout »)	134 (1.2%)	82.8%	154 (1.4%)	81.8%
Tous les cas	11 024 (100%)	95.6%	11 069 (100%)	95.3%

On constate en premier lieu, comme on pouvait s'y attendre, qu'une performance plus élevée est obtenue en se limitant aux homographes ne contenant que deux classes possibles, en comparaison avec le résultat global incluant tous les cas. Mais cette différence n'est pas grande, considérant que les homographes avec deux classes grammaticales comptent pour plus de 92% des homographes. Comme prévu, on constate aussi une baisse de la performance quand on s'attarde aux homographes comportant trois classes possibles. Une baisse assez notable, surtout dans le cas du roman de science-fiction (chute d'environ 96% à 88%). On se rappellera qu'il n'y a, au sein des deux corpus, qu'un seul homographe comportant quatre classes possibles, soit l'homographe « point ». La performance de désambiguïsation pour ce cas précis est très élevée, car comme on l'a vu à la Section 5.3.2.5, le test de locutions (non basé sur l'apprentissage machine) a été très efficace pour cet homographe. On ne compte au sein des deux corpus qu'un seul homographe aussi, dans le cas à cinq classes grammaticales (« tout »). Dans ce cas par contre, l'efficacité est bien moindre (à peine au-dessus de 80%). On peut donc en conclure, comme on s'y attendait, qu'à l'exception du cas « point », l'efficacité de la désambiguïsation décroît avec le nombre de classes grammaticales possibles. Mais comme les cas avec plus de deux classes grammaticales demeurent peu fréquents en comparaison avec les cas avec deux classes, la réduction globale de la performance leur étant due n'est pas marquée.

5.3.2.8. Performance de la méthode avec phrase complète

Au Chapitre 4, deux approches de désambiguïsation au niveau de la phrase ont été décrites : l'approche dite « gauche-droite » (Section 4.6.4.3) et l'approche « globale » basée sur la phrase complète (Section 4.6.4.4). Avec l'approche « gauche-droite », on désambiguïse chaque phrase un mot à la fois, avançant tout simplement de gauche à droite. Il en découle que les homographes situés à la droite du mot sous étude ne sont pas encore désambiguïsés, ce qui introduit un élément d'incertitude au moment de l'évaluation des caractéristiques d'apprentissage machine. La méthode « globale » (ou « phrase complète) a été mise au point pour pallier cette faiblesse. Avec cette approche, on tente d'évaluer toutes les combinaisons possibles de classes grammaticales pour tous les homographes de la phrase. On peut ainsi évaluer les caractéristiques de tous les homographes de façon plus appropriée, n'ayant plus à se préoccuper de l'incertitude associée à des homographes non encore désambiguïsés. Il avait finalement fallu mettre au point un système de pointage pour identifier la combinaison la plus probable de classes grammaticales pour la phrase dans son ensemble. On se contente ici de comparer la performance de désambiguïsation selon ces deux approches.

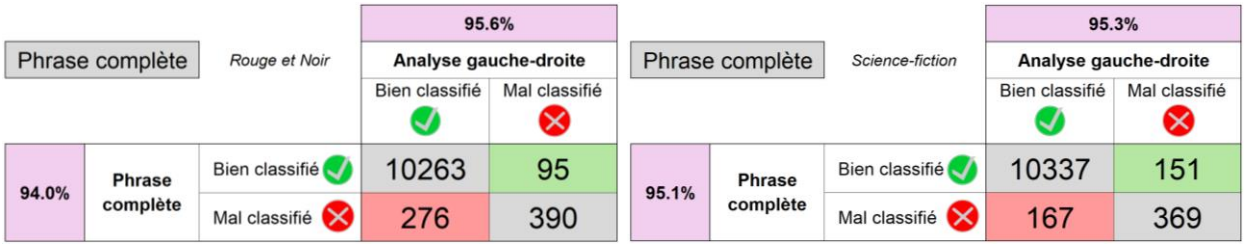


Figure 5.20 : Matrices de contribution pour *tous les homographes du corpus*, illustrant la performance de désambiguïsation en utilisant l'analyse gauche-droite ou l'analyse basée sur la phrase complète, pour le roman « Le Rouge et le Noir » (à gauche) et le roman de science-fiction (à droite)

Une fois de plus, on a recours à des matrices de contribution (Figure 5.20), une pour chacun des deux corpus, pour comparer les deux approches. On y quantifie la « contribution » de l'analyse de la phrase complète à l'amélioration de la désambiguïsation, en comparaison avec le cas de l'analyse dite « gauche-droite ». Tout comme précédemment, on retrouve dans les cases vertes les cas où l'analyse basée sur la phrase complète a permis de correctement classifier des homographes qui ne l'étaient pas précédemment, sur la base de l'analyse « gauche-droite ». De la même façon, on retrouve au contraire dans les cases rouges les cas où l'analyse basée sur la phrase complète a incorrectement classifié des homographes qui étaient précédemment classifiés, sur la base de l'analyse « gauche-droite ».

On constate hélas que pour les deux corpus, l'analyse basée sur la phrase complète n'a pas donné les résultats escomptés. En effet, dans les deux cas, la performance globale de désambiguïsation s'est trouvée *réduite* en comparaison avec l'analyse « gauche-droite ». On retrouve bien sûr un certain nombre d'homographes qui ont bénéficié de l'approche basée sur la phrase complète (95 pour « Le Rouge et le Noir » et 151 pour le roman de science-fiction). Mais on retrouve en plus grand nombre des cas où l'analyse « gauche-droite » s'était avérée supérieure (276 pour « Le Rouge et le Noir » et 167 pour le roman de science-fiction). On retrouve des exemples précis issus du roman « Le Rouge et le Noir » à l'Annexe G. La performance globale de désambiguïsation ne s'est toutefois pas retrouvée à être trop affectée. On constate en effet une chute d'uniquement 1.6% pour le roman « Le Rouge et le Noir » (de 95.6% à 94.0%) et une chute encore plus faible de 0.2% (de 95.3% à 95.1%) dans le cas du roman de science-fiction.

L'explication la plus plausible pour la plus grande efficacité de l'approche « gauche-droite » est probablement que justement, les locuteurs et scripteurs de la langue française bâtissent leurs phrases en procédant de gauche à droite. Un nouveau mot se trouve inséré à la phrase avant même, dans bien des cas, que les mots suivants soient encore « prévus » par le locuteur. Les mots subséquents s'ajustent donc en fonction de ce qui a été dit ou écrit précédemment. Il se peut aussi que l'approche basée sur la phrase complète ait pu bénéficier d'un meilleur algorithme pour le pointage de chaque phrase. À la Figure 4.31, on illustrait par exemple le fait que le score puisse se calculer sur la base d'une multiplication ou d'une addition des probabilités pour chacun des homographes de la phrase. Mais une analyse sommaire (non illustrée ici) n'a pu démontrer de différence notable entre ces deux approches (addition vs. multiplication des probabilités).

Une étude plus approfondie de la méthode d'analyse basée sur la phrase complète déborde de la portée de ce projet. Pour le restant de ce mémoire, c'est donc la méthode d'analyse « gauche-droite » qui sera appliquée, sur la base de sa meilleure efficacité. De plus, la méthode gauche-droite a l'avantage d'une exécution plus rapide, comme on le démontrera à la Section 5.5.

5.3.3. Matrices de confusion finales

À cette étape, tous les algorithmes contribuant à l'algorithme global de désambiguïsation ont été décrits, inclus et validés. Un seul d'entre eux, l'analyse de la phrase complète, s'est avéré peu judicieux et n'a donc pas été maintenu. On utilise donc l'algorithme dit « gauche-droite » pour l'analyse des phrases. En revanche, les « tests spécialisés » (Section 5.3.2.3), les tests de participes passés (Section 5.3.2.4), les tests de locutions (Section 5.3.2.5) et les tests statistiques (Section 5.3.2.6) ont été maintenus dans l'algorithme final. On peut donc maintenant présenter les matrices de confusion finales pour les deux corpus, basées sur la désambiguïsation faisant appel aux algorithmes mentionnés plus haut.

Pour le roman « Le Rouge et le Noir », la Figure 5.21 fournit la matrice de confusion globale, impliquant les neuf classes grammaticales utilisées pour ce projet. On y retrouve la performance globale de 95.6% (case mauve) obtenue plus haut à la Figure 5.19. On constate aussi évidemment que les cases vertes (bonnes classifications) contiennent des valeurs bien plus élevées que les cases rouges (mauvaises classifications). On observe aussi un bon nombre de cases blanches. Celles-ci correspondent aux paires de classes grammaticales ne correspondant à aucun homographe observé dans le roman. Par exemple, on ne compte aucun homographe de type « nom vs. déterminant ». Les chiffres les plus élevés des cases rouges correspondent aux paires « conjonction vs. pronom » (88) et « déterminant vs. préposition » (82). Mais une étude approfondie de paires en particulier s'effectue de façon plus efficace en générant des matrices de confusion pour chaque paire possible de classes grammaticales. C'est ce qu'on a illustré à la Figure 5.22. On y retrouve en effet 24 matrices de confusion (sur un total maximum de 36).

La plus faible performance obtenue est pour la paire « adverbe vs. pronom » (83.1%). On constate qu'uniquement 40% des prédictions de pronoms se sont avérées justes dans ce cas. En fait, le seul homographe dans cette catégorie est le mot « tout », qui, comme on l'a décrit précédemment, peut aussi être un adjectif, un nom ou un adverbe. Il n'est donc pas surprenant que la performance pour cette paire corresponde à la performance de désambiguïsation du mot « tout » dans le roman « Le Rouge et le Noir »[11].

Une autre paire de classes grammaticales avec faible performance est la paire « conjonction vs. pronom », avec une performance de 86.8%. On avait d'ailleurs noté plus haut que cette paire avait mené au plus grand nombre de mauvaises classifications dans la matrice de confusion globale de la Figure 5.21. Dans cette catégorie, on retrouve les homographes « que », « qu' » et « s' ». La performance pour ces homographes avait été discutée dans le cadre des tests spécialisés (Section 5.3.2.3). Un test de locution avait aussi été inclus pour « que » et « qu' ». Tandis que la désambiguïsation s'est avérée parfaite pour l'homographe « s' », ce n'est pas le cas pour les homographes « que » et « qu' », qui présentent un grand défi.

La seule autre paire de classes grammaticales ayant mené à une performance sous 90% est « nom vs. pronom » (88.0%). Mais ces cas sont si peu nombreux, que seuls 3 homographes de ce type ont été mal classifiés dans la portion lemmatisée du roman. La paire « déterminant vs. préposition », responsable de 82 erreurs de classification, tel que mentionné plus haut, correspond quant à elle à une performance de 96.2%, donc supérieure à la performance globale, toutes classes grammaticales confondues. Le grand nombre d'erreurs est simplement causé par la très grande fréquence d'homographes de ce type (les homographes « de » et « d' »).

[11] Il est à noter que la performance de désambiguïsation pour le mot « tout » affichée au Tableau 5.14 (82.8%) avait été obtenue *avant* l'application de certains autres tests (participes passés, locutions, statistiques), ce qui explique la légère différence avec le résultat de 83.1% illustré à la Figure 5.22

Classes grammaticales réelles \ Classes grammaticales prédites par l'algorithme	Verbe	Adj	Nom	Adv	Dét	Pron	Prép	Conj	Inter	Rappel
Verbe	1061	30	35							94.2%
Adj	23	1137	47	4	2					93.7%
Nom	15	34	1514	3		1	2	1		96.4%
Adv	5	6	3	772	7	9				96.3%
Dét		3	7	1	2611	8	82			96.3%
Pron			2	1	19	753		39		92.5%
Prép	2		5				2187			99.7%
Conj			1			88		498		84.8%
Inter									5	100.0%
Précision	95.9%	94.0%	93.8%	98.8%	98.9%	87.7%	96.3%	92.6%	100.0%	95.6%

Figure 5.21 : Matrice de confusion globale pour le roman « Le Rouge et le Noir »

	Verbe	Adjectif	
Verbe	639	30	95.5%
Adjectif	23	655	96.6%
	96.5%	95.6%	**96.1%**

	Verbe	Nom	
Verbe	573	35	94.2%
Nom	15	1223	98.8%
	97.4%	97.2%	**97.3%**

	Verbe	Adverbe	
Verbe	0	0	n/a
Adverbe	5	236	97.9%
	0.0%	100.0%	**97.9%**

	Verbe	Pronom	
Verbe	2	0	100.0%
Pronom	0	0	n/a
	100.0%	n/a	**100.0%**

	Verbe	Préposition	
Verbe	2	0	100.0%
Préposition	2	67	97.1%
	50.0%	100.0%	**97.2%**

	Verbe	Conjonction	
Verbe	0	0	n/a
Conjonction	0	1	100.0%
	n/a	100.0%	**100.0%**

	Verbe	Interjection	
Verbe	0	0	n/a
Interjection	0	2	100.0%
	n/a	100.0%	**100.0%**

	Adjectif	Nom	
Adjectif	487	47	91.2%
Nom	34	515	93.8%
	93.5%	91.6%	**92.5%**

	Adjectif	Adverbe	
Adjectif	46	4	92.0%
Adverbe	6	167	96.5%
	88.5%	97.7%	**95.5%**

	Adjectif	Déterm.	
Adjectif	5	2	71.4%
Déterm.	3	69	95.8%
	62.5%	97.2%	**93.7%**

	Adjectif	Pronom	
Adjectif	22	0	100.0%
Pronom	0	21	100.0%
	100.0%	100.0%	**100.0%**

	Adjectif	Conjonction	
Adjectif	0	0	n/a
Conjonction	0	1	100.0%
	n/a	100.0%	**100.0%**

	Nom	Adverbe	
Nom	62	3	95.4%
Adverbe	3	433	99.3%
	95.4%	99.3%	**98.8%**

	Nom	Déterm.	
Nom	24	0	100.0%
Déterm.	7	350	98.0%
	77.4%	100.0%	**98.2%**

	Nom	Pronom	
Nom	9	1	90.0%
Pronom	2	13	86.7%
	81.8%	92.9%	**88.0%**

	Nom	Préposition	
Nom	16	2	88.9%
Préposition	5	58	92.1%
	76.2%	96.7%	**91.4%**

	Nom	Conjonction	
Nom	4	1	80.0%
Conjonction	1	23	95.8%
	80.0%	95.8%	**93.1%**

	Nom	Interjection	
Nom	5	0	100.0%
Interjection	0	0	n/a
	100.0%	n/a	**100.0%**

	Adverbe	Déterm.	
Adverbe	43	7	86.0%
Déterm.	1	54	98.2%
	97.7%	88.5%	**92.4%**

	Adverbe	Pronom	
Adverbe	43	9	82.7%
Pronom	1	6	85.7%
	97.7%	40.0%	**83.1%**

	Adverbe	Conjonction	
Adverbe	92	0	100.0%
Conjonction	0	53	100.0%
	100.0%	100.0%	**100.0%**

	Déterm.	Pronom	
Déterm.	2308	8	99.7%
Pronom	19	316	94.3%
	99.2%	97.5%	**99.0%**

	Déterm.	Préposition	
Déterm.	0	82	0.0%
Préposition	0	2071	100.0%
	n/a	96.2%	**96.2%**

	Pronom	Conjonction	
Pronom	415	39	91.4%
Conjonction	88	421	82.7%
	82.5%	91.5%	**86.8%**

Figure 5.22 : Matrices de confusion pour toutes les paires de classes grammaticales pour lesquelles il existe au moins un homographe, dans le roman « Le Rouge et le Noir »

Classes grammaticales réelles	Classes grammaticales prédites par l'algorithme: Verbe	Adj	Nom	Adv	Dét	Pron	Prép	Conj	Inter	Rappel
Verbe	1452	36	72	1		1				93.0%
Adj	27	1097	40	9	1	1				93.4%
Nom	24	36	1627	6	5	2			6	95.4%
Adv		14	9	867	4	3		3		96.3%
Dét		1	4	3	2721	17	38			97.7%
Pron		1	2	10	39	622		52		85.7%
Prép	8		11		9		1715			98.4%
Conj				2		28		433		93.5%
Inter			1						8	88.9%
Précision	96.1%	92.6%	92.1%	96.5%	97.9%	92.3%	97.8%	88.7%	57.1%	**95.3%**

Figure 5.23 : Matrice de confusion globale pour le roman de science-fiction

	Verbe	Adjectif	
Verbe	567	36	94.0%
Adjectif	27	538	95.2%
	95.5%	93.7%	**94.6%**

	Verbe	Nom	
Verbe	1062	72	93.7%
Nom	24	1306	98.2%
	97.8%	94.8%	**96.1%**

	Verbe	Adverbe	
Verbe	0	1	0.0%
Adverbe	0	238	100.0%
	n/a	99.6%	**99.6%**

	Verbe	Pronom	
Verbe	2	1	66.7%
Pronom	0	0	n/a
	100.0%	0.0%	**66.7%**

	Verbe	Préposition	
Verbe	0	0	n/a
Préposition	8	79	90.8%
	0.0%	100.0%	**90.8%**

	Verbe	Conjonction	
Verbe	0	0	n/a
Conjonction	0	1	100.0%
	n/a	100.0%	**100.0%**

	Verbe	Interjection	
Verbe	0	0	n/a
Interjection	0	0	n/a
	n/a	n/a	**n/a**

	Adjectif	Nom	
Adjectif	479	40	92.3%
Nom	36	510	93.4%
	93.0%	92.7%	**92.9%**

	Adjectif	Adverbe	
Adjectif	44	9	83.0%
Adverbe	14	120	89.6%
	75.9%	93.0%	**87.7%**

	Adjectif	Déterm.	
Adjectif	4	1	80.0%
Déterm.	1	48	98.0%
	80.0%	98.0%	**96.3%**

	Adjectif	Pronom	
Adjectif	89	1	98.9%
Pronom	1	75	98.7%
	98.9%	98.7%	**98.8%**

	Adjectif	Conjonction	
Adjectif	0	0	n/a
Conjonction	0	0	n/a
	n/a	n/a	**n/a**

	Nom	Adverbe	
Nom	110	6	94.8%
Adverbe	9	538	98.4%
	92.4%	98.9%	**97.7%**

	Nom	Déterm.	
Nom	16	5	76.2%
Déterm.	4	363	98.9%
	80.0%	98.6%	**97.7%**

	Nom	Pronom	
Nom	1	2	33.3%
Pronom	2	49	96.1%
	33.3%	96.1%	**92.6%**

	Nom	Préposition	
Nom	1	0	100.0%
Préposition	11	144	92.9%
	8.3%	100.0%	**92.9%**

	Nom	Conjonction	
Nom	1	1	50.0%
Conjonction	0	4	100.0%
	100.0%	80.0%	**83.3%**

	Nom	Interjection	
Nom	2	6	25.0%
Interjection	1	8	88.9%
	66.7%	57.1%	**58.8%**

	Adverbe	Déterm.	
Adverbe	47	4	92.2%
Déterm.	3	41	93.2%
	94.0%	91.1%	**92.6%**

	Adverbe	Pronom	
Adverbe	47	3	94.0%
Pronom	10	38	79.2%
	82.5%	92.7%	**86.7%**

	Adverbe	Conjonction	
Adverbe	33	3	91.7%
Conjonction	2	37	94.9%
	94.3%	92.5%	**93.3%**

	Déterm.	Pronom	
Déterm.	2371	17	99.3%
Pronom	39	268	87.3%
	98.4%	94.0%	**97.9%**

	Déterm.	Préposition	
Déterm.	23	38	37.7%
Préposition	9	1518	99.4%
	71.9%	97.6%	**97.0%**

	Pronom	Conjonction	
Pronom	306	52	85.5%
Conjonction	28	392	93.3%
	91.6%	88.3%	**89.7%**

Figure 5.24 : Matrices de confusion pour toutes les paires de classes grammaticales pour lesquelles il existe au moins un homographe, dans le roman de science-fiction. Les paires « adjectif déterminant » et « adjectif conjonction », même si vides, sont affichées pour comparaison directe avec le cas du roman « Le Rouge et le Noir »

La Figure 5.23 fournit la matrice de confusion globale pour le roman de science-fiction. On y retrouve la performance globale de 95.3% (case mauve) obtenue plus haut à la Figure 5.19. On constate aussi évidemment que les cases vertes (bonnes classifications) contiennent des valeurs bien plus élevées que les cases rouges (mauvaises classifications). On observe aussi que certaines cases qui étaient rouges dans le cas du roman « Le Rouge et le Noir » sont devenues blanches dans le cas du roman de science-fiction, et vice-versa. Cela dépend des homographes précis présents ou non dans ces deux corpus, pour les paires de classes grammaticales qui sont peu fréquentes.

Le chiffre le plus élevé parmi les cases rouges correspond à la paire « verbe vs. nom » (72). Cette observation diffère de ce qu'on avait vu avec le roman « Le Rouge et le Noir » où elles étaient proportionnellement moins nombreuses. La présence plus nombreuse d'erreurs de classification pour cette paire dans le roman de science-fiction s'explique aisément. Le roman de science-fiction étant écrit principalement au présent de l'indicatif, on y retrouve davantage d'homographes comme « demande », « contrôle », etc. qui correspondent à des formes verbales au présent de l'indicatif. En revanche, le roman « Le Rouge et le Noir » a fait davantage appel au passé simple et à l'imparfait, qui ne génèrent pas autant d'homographes de ce type. Le chiffre le plus élevé suivant (52) correspond à la paire « conjonction vs. pronom », qui avait été identifiée comme étant la plus nombreuse pour le roman « Le Rouge et le Noir ».

Là encore, il est plus intéressant de s'attarder à des paires de classes grammaticales en particulier en observant les matrices de confusion pour chaque paire possible de classes grammaticales. C'est ce qu'on a illustré à la Figure 5.24. Le plus faible résultat s'obtient cette fois pour la paire « nom vs. interjection » (58.8%), qui contient par exemple l'homographe « merci ». Mais ces cas étant si peu nombreux, ils ne résultent qu'en seulement 7 homographes mal classifiés globalement. On voit ensuite une performance de 66.7% pour la paire « verbe vs. pronom », mais ces cas sont encore moins nombreux, étant responsables d'uniquement un homographe mal classé (l'homographe « tiens »). Il en est de même pour la paire « nom vs. conjonction », avec un score de 83.3% mais un seul homographe mal classifié (« or »). On suit ensuite avec la paire « adverbe vs. pronom » (86.7%, pour le mot « tout »), « adjectif vs. adverbe » (87.7%, « vite », « même », « bizarre » et « soudain ») et finalement « pronom vs. conjonction » (89.7%), correspondant aux homographes « que », « qu' » et « s' », tel que discuté pour le roman « Le Rouge et le Noir ». Toutes les autres paires ont mené à des performances excédant 90%.

On note finalement que deux matrices de confusion de la Figure 5.24 sont vides, correspondant aux paires « adjectif vs. déterminant » et « adjectif vs. conjonction ». Celles-ci ont été maintenues dans cette figure uniquement car elles apparaissaient à la Figure 5.22 pour le roman « Le Rouge et le Noir », et donc pour faciliter la comparaison directe entre ces deux figures.

5.3.4. Analyse et groupement des erreurs de désambiguïsation

Tel que vu précédemment, une performance de désambiguïsation de 95.6% a été obtenue pour le roman « Le Rouge et le Noir », à la suite de l'analyse des 11 024 premiers homographes de ce roman. Bien que cette performance soit excellente, il n'en demeure pas moins que 485 homographes ont été mal classifiés. Dans cette section, nous chercherons à déterminer les causes générales pour lesquelles ces homographes ont résisté à tous les algorithmes et tests mis en place pour ce projet. Une telle analyse comporte deux buts : identifier le potentiel d'amélioration de l'algorithme de désambiguïsation actuel, mais aussi identifier les plus grands défis auxquels fait face un lemmatiseur, dans le but d'évaluer l'efficacité d'outils de lemmatisation existants lorsqu'on les confronte à des phrases qu'on sait qu'elles posent problème.

Tableau 5.19 : Liste de tous les homographes mal classifiés du roman « Le Rouge et le Noir », parmi les 11024 premiers, avec leurs fréquences. Seuls les cas apparaissant au moins deux fois sont listés ici

Homographe	Fréquence	Homographe	Fréquence	Homographe	Fréquence
de	65	dessous	3	entre	2
que	64	dire	3	fait	2
qu'	63	domestique	3	les	2
tout	23	fou	3	leur	2
dit	18	haut	3	libéral	2
d'	17	neuf	3	livres	2
le	12	nouveau	3	mépris	2
vieux	8	paysan	3	mort	2
l'	6	vite	3	nuit	2
son	6	allée	2	parti	2
être	5	assis	2	sourire	2
bas	4	complet	2	vers	2
plus	4	dîner	2	voyant	2
bien	3	dites	2		

L'analyse ici se limite au roman « Le Rouge et le Noir », car on émet l'hypothèse que les sources d'erreurs générales seraient semblables pour le roman de science-fiction. Parmi les 485 homographes mal classifiés dans la portion lemmatisée de ce roman, on ne retrouve en fait que 165 entrées distinctes. Le Tableau 5.19 dresse la liste des 41 cas parmi ceux-ci apparaissant au moins deux fois.

On constate que les quatre premiers homographes de la liste avaient eu droit à leur propre « test spécialisé » (Section 5.3.2.3). C'est donc dire que le défi de désambiguïsation relié à ces homographes en particulier a déjà été reconnu, et que malgré ces tests spécialisés, ces homographes sont encore ceux qui causent le plus de problèmes. On constate aussi que plusieurs des autres mots avaient aussi été traités par les tests spécialisés (« le », « l' », « bien ») ou des tests de locutions (« être », « fait »). Plusieurs autres sont des participes passés (« dit », « allée », « assis », « fait », « mépris », « mort », « parti »), qui ont donc aussi été traités par un test additionnel. Mais on retrouve tout de même ces homographes dans cette liste d'erreurs. On cherche donc ici à comprendre pourquoi ces mots, malgré tous ces tests, ont été mal classifiés.

À chacun des 485 homographes mal classifiés, on a assigné une ou plusieurs « causes » plus ou moins génériques pour cette mauvaise assignation. Il ne s'agit pas ici d'un exercice à prétention « scientifique », mais qui aide tout de même à donner une idée d'ensemble des difficultés de la désambiguïsation rencontrées pendant ce projet. Puisqu'on peut assigner plus d'une cause à chaque homographe mal classifié, les résultats, affichés au Tableau 5.20, sont exprimés en pourcentage, où la somme des pourcentages donne 100%. Voici une brève explication des quelques causes génériques auxquelles ont été associés les homographes mal classifiés :

- **Cas limite (presque réussi) :** la probabilité de la classe grammaticale prédite était uniquement de peu supérieure au seuil de 0.5 (jusqu'à 0.6)
- **Structure de phrase inhabituelle :** homographe au sein d'une phrase dont la structure n'est pas commune, par exemple un nom commun non introduit par un déterminant, ou une phrase qui commence par « que »
- **Phrase complexe :** cas qui demanderaient que l'algorithme interprète le sens de la phrase pour arriver à bien choisir la bonne classe grammaticale
- **Homographes environnants mal désambiguïsés :** une mauvaise classification résultant d'un autre homographe en amont mal désambiguïsé, ou encore de la considération de classes non appropriées pour les homographes en aval non encore désambiguïsés
- **Aurait bénéficié de l'analyse phrase complète :** cas où la désambiguïsation a été réussie avec l'algorithme de phrase complète
- **Algorithme statistique de verbes n'a pas bien fonctionné :** cas où un homographe a été erronément considéré comme un verbe, malgré le test statistique
- **Algorithme pour participes passés n'a pas bien fonctionné :** l'algorithme de participes passés a mené à une mauvaise piste
- **Pourrait être bien classifié avec ajout d'une locution :** l'homographe pourrait être correctement classifié si on ajoutait la locution appropriée à la « banque de données » des locutions
- **Adjectif antéposé interprété comme un nom commun :** certains adjectifs peuvent être utilisés avant ou après le nom qu'ils modifient, tout en étant potentiellement aussi des noms communs, par exemple « vieux »
- **Cas particulier "de" comme déterminant :** ce déterminant peut être utilisé dans certains cas précis, comme « pas de place » impliquant le négatif, ou encore dans certains cas au pluriel (« ce jardin comprend de belles fleurs ») comme substitut au déterminant « des »

Tableau 5.20 : Liste et fréquences relatives des causes de mauvaises classifications des homographes dans la portion lemmatisée du roman « Le Rouge et le Noir » (485 erreurs). Plus d'une cause peut être assignée à chaque cas

Causes de mauvaise classification	Fréquence relative
Cas limite	7.4%
Structure inhabituelle	12%
Phrase complexe	21%
Homographes environnants mal désambiguïsés	2.1%
Aurait bénéficié de l'analyse phrase complète	1.1%
Test statistique de verbes n'a pas bien fonctionné	15%
Test de participes passés n'a pas bien fonctionné	9.6%
On pourrait ajouter une autre locution	13%
Adjectif antéposé interprété comme un nom	3.2%
Cas particulier du déterminant « de »	16%

À la lecture du Tableau 5.20, on se rend compte que le pourcentage le plus élevé correspond au cas « phrase complexe ». Hélas, il n'y a pas de solution facile pour de tels cas. Il sera intéressant de confronter (au Chapitre 6) des outils existants de lemmatisation à de tels cas de phrases complexes. En revanche, une autre valeur assez élevée concerne l'homographe « de », quand celui-ci est un déterminant. Un test spécialisé avait pourtant été mis en place, avec des caractéristiques semblant appropriées. Mais dans ce cas précis, la faiblesse de désambiguïsation est probablement due au fait qu'en comparaison avec les cas où l'homographe « de » est une préposition, les cas de déterminants sont assez rares. Un tel débalancement peut affecter l'efficacité de l'apprentissage machine. Différentes stratégies pourraient être mises en place pour s'attaquer à ce problème de débalancement.

Dans plusieurs cas, de nouvelles locutions pourraient être ajoutées à la banque de données, afin de faciliter la désambiguïsation. Mais l'ajout de locutions est une tâche ardue, car celles-ci sont nombreuses, et peuvent se limiter à ne corriger qu'un faible nombre d'homographes dans chaque cas. C'est là qu'un travail d'équipe regroupant les travaux de plusieurs chercheurs pourrait porter fruit, afin de comptabiliser le plus grand nombre possible de locutions.

On retrouve aussi au Tableau 5.20 un bon pourcentage de cas de structures de phrases inhabituelles. Ces structures pourraient être traitées à la manière des locutions, en insérant des tests particuliers pour chacune de ces structures peu fréquentes. Là encore, une telle approche demande un grand effort pour identifier un grand nombre de structures pertinentes.

Finalement, on note aussi un grand nombre de « cas limite ». On ne peut qu'espérer qu'avec de plus grandes banques de données et l'analyse de corpus lemmatisés manuellement plus volumineux, on arrivera éventuellement à de meilleures prédictions de l'apprentissage machine.

À l'Annexe H, on fournit plusieurs exemples d'homographes mal classifiés en référence, listés selon les différentes causes génériques apparaissant au Tableau 5.20.

5.3.5. Statistiques globales du corpus après désambiguïsation

L'objectif de l'Étape 1 est de générer l'information nécessaire à l'algorithme de l'Étape 2, afin que les phrases générées aléatoirement soient le plus possible représentatives du corpus de référence. En ce sens, il est utile d'étudier les statistiques globales de ce corpus, pour pouvoir ensuite comparer ces statistiques avec celles du texte généré aléatoirement à l'Étape 2. De plus, cette analyse permet de se pencher davantage sur le concept d'homographes, et en particulier, d'en quantifier l'ampleur dans le texte de référence choisi.

La présente section revisite donc les statistiques globales sur le corpus de référence, après désambiguïsation. Il importe toutefois de se rappeler que ce processus n'est pas parfait, sa performance globale ayant été évaluée à environ 95% à la Section 5.3 pour les deux corpus de référence, en utilisant la méthode *k-fold*. Tout de même, les statistiques après désambiguïsation devraient être bien plus représentatives du corpus de référence, que celles obtenues avant ce processus.

5.3.5.1. Longueur des phrases

Après exécution de l'algorithme de lemmatisation appliqué au corpus de référence, on constate que le roman « Le Rouge et le Noir » contient 10 639 phrases, comprenant 182 205 mots. La Figure 5.25 fournit la distribution de la longueur des phrases, sous forme d'histogramme. On y affiche la moyenne du nombre de mots par phrase (17) ainsi que la médiane (22). La Figure 5.26 fournit la même information, mais cette fois pour le roman de science-fiction. Cette fois, on obtient une moyenne de 10 mots par phrase, et une médiane de 13 On constate donc que le roman « Le Rouge et le Noir » comprend des phrases beaucoup plus longues en moyenne que le roman de science-fiction.

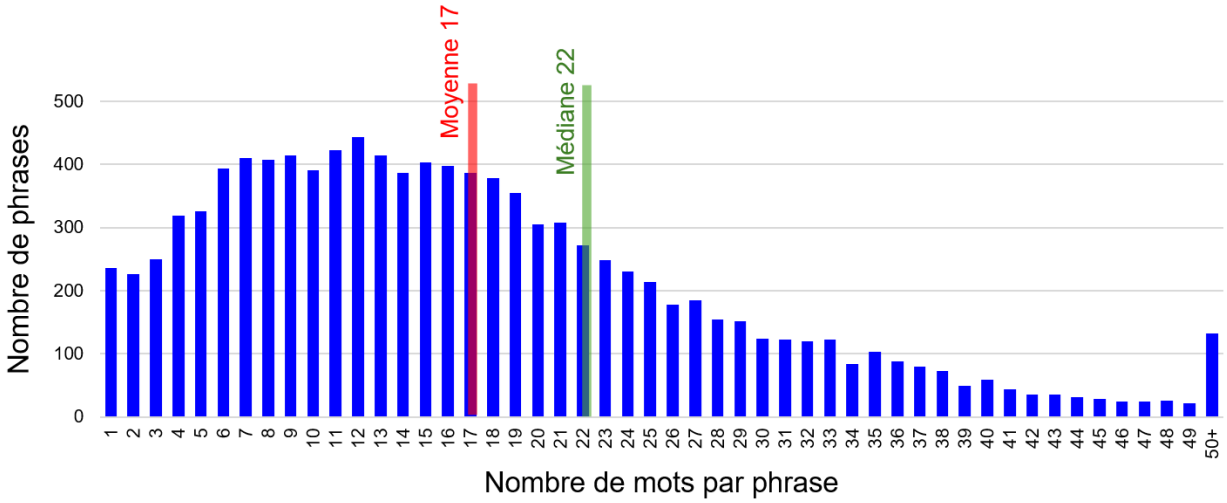


Figure 5.25 : Histogramme de la longueur des phrases dans le roman « Le Rouge et le Noir »

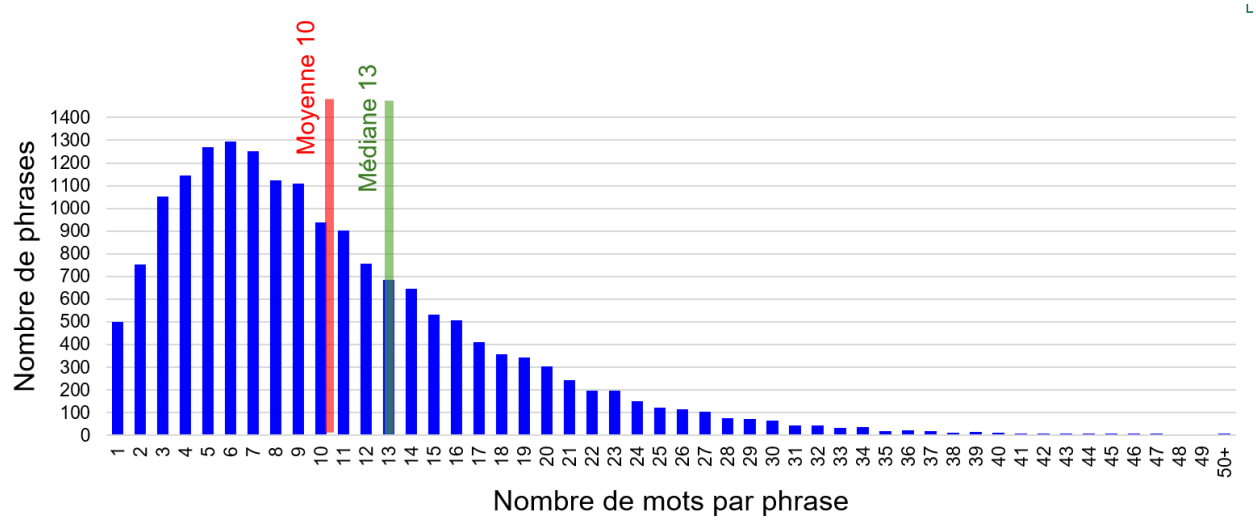


Figure 5.26 : Histogramme de la longueur des phrases dans le roman de science-fiction

5.3.5.2. Lemmes les plus fréquents du corpus

Le premier objectif d'un lemmatiseur est d'associer chaque mot à son lemme. Une fois cette opération effectuée, on peut aisément regrouper tous les mots partageant le même lemme, et ensuite étudier les statistiques en lien avec les lemmes. Le Tableau 5.21 permet justement une telle analyse, en fournissant les fréquences d'occurrence des principaux lemmes des deux corpus de référence, pour chacune des neuf classes grammaticales principales.

À la suite de la désambiguïsation effectuée à l'Étape 1, la plupart des mots du corpus sont désormais associés à un seul lemme. Seuls les homographes issus d'une seule classe grammaticale font exception. Par exemple, l'homographe « suis » demeure associé à deux lemmes, soit les verbes « être » et « suivre ». Il est tout de même intéressant de se pencher ici sur les statistiques concernant les lemmes distincts rencontrés dans le corpus de référence. Le nombre total de lemmes distincts est fourni au Tableau 5.21. On y retrouve non seulement le

nombre total de lemmes distincts, mais aussi leurs regroupements par classe grammaticale. On ne tient pas compte dans ce tableau de la fréquence d'apparition des lemmes en question, simplement de leur présence en au moins un exemplaire. Il est à noter que les noms propres ont été exclus de cette analyse.

On constate que les pourcentages de lemmes distincts par classe grammaticale sont assez semblables pour les deux corpus de référence, malgré le style distinct des deux auteurs. On peut donc émettre l'hypothèse que les proportions observées sont possiblement assez générales pour des textes de langue française.

L'information au Tableau 5.21 nous permet d'évaluer de façon sommaire la richesse lexicale des deux corpus de référence, qu'on peut définir comme le ratio du nombre de lemmes distincts sur le nombre total de mots. On obtient donc ici un ratio de 0.035 (6239 divisé par 176 175) pour le roman « Le Rouge et le Noir » et un ratio de 0.044 pour le roman de science-fiction (7798 divisé par 175 712). On constate donc que le nombre de lemmes distincts utilisés dans les corpus de référence est faible, en comparaison avec le lexique global de la langue française, qui comprend plusieurs dizaines de milliers de mots. Mais c'est là une caractéristique de la plupart des textes de langue française. Il est rare en effet, qu'on y retrouve plus que quelques milliers de lemmes distincts, même dans le cas d'œuvres littéraires d'auteurs classiques (Muller, 1971).

Au Tableau 5.22, on s'attarde maintenant aux lemmes les plus fréquents du roman « Le Rouge et le Noir », en fonction de leur classe grammaticale. Le Tableau 5.23 offre la même information, mais cette fois pour le roman de science-fiction. On y constate que les 7 verbes les plus courants du roman « Le Rouge et le Noir » et les 10 verbes les plus courants du roman de science-fiction sont des verbes du troisième groupe.

Tableau 5.21 : Nombre total de lemmes distincts dans les deux corpus de référence, groupés par classes grammaticales (ne tenant pas compte de leur fréquence d'apparition). Noms propres et mots non classifiés non compris

	Le Rouge et le Noir		Science-fiction	
Classe grammaticale	Nombre de lemmes distincts	Pourcentage du total	Nombre de lemmes distincts	Pourcentage du total
Verbes	1233	20%	1824	23%
Adjectifs	1660	27%	2089	27%
Noms	2895	46%	346	43%
Adverbes	288	4.6%	364	4.7%
Déterminants	50	0.80%	42	0.54%
Pronoms	43	0.69%	47	0.60%
Prépositions	35	0.56%	35	0.45%
Conjonctions	24	0.38%	25	0.32%
Interjections	11	0.18%	26	0.33%
TOTAL	6239	100%	7798	100%

Bien que la grande majorité des verbes de la langue française soit du premier groupe, ceux qui sont utilisés les plus fréquemment sont effectivement du troisième groupe. Ceux-ci sont caractérisés par leur irrégularité. Cette prévalence des verbes du troisième groupe ainsi que leur irrégularité confirment la pertinence d'une opération de lemmatisation détaillée. En effet, une technique de lemmatisation plus simple basée sur la troncation ne serait pas très efficace pour des verbes très irréguliers. Par exemple, le verbe « être » comporte des graphies très différentes, telles que « suis », « es », « sommes », « êtes », « serez », etc.

On constate aussi que parmi les 10 lemmes les plus courants de chaque classe grammaticale, on retrouve souvent les mêmes au sein des deux corpus. En effet, 6 verbes, 8 adjectifs, 6 adverbes, 9 déterminants, 8 pronoms, et 9 prépositions sont communs aux deux corpus sur les 10 les plus courants. On peut émettre l'hypothèse que dans la plupart des textes de langue française, on retrouverait des lemmes semblables parmi les plus courants pour ces classes grammaticales.

Les deux corpus diffèrent en revanche au niveau des noms et des interjections. On ne retrouve en effet qu'un seul nom commun aux deux listes (œil), et seules trois interjections communes (ah, eh, et hélas). C'est donc au sein de ces deux classes grammaticales (noms et interjections) qu'on peut retrouver le plus d'indices du sujet ou thème du corpus de référence. En particulier, pour le roman de science-fiction, on retrouve les lemmes « terriens » et « vaisseau », qui ne figurent même pas dans la liste de lemmes du roman « Le Rouge et le Noir » Ce dernier contient quant à lui des interjections représentatives du lieu et de l'époque où se déroule l'action du roman, soit « parbleu » et « morbleu ».

Les Figures 5.27 et 5.28 quant à elles, laissent de côté les classes grammaticales et indiquent les fréquences d'apparition des 20 lemmes les plus courants des deux corpus de référence. On y surligne en jaune les mots parmi ceux-ci qui sont associés à plus d'une classe grammaticale, donc des homographes. On constate que la plupart de ces 20 lemmes sont associés à des homographes, ce qui, une fois de plus, met en lumière la problématique des homographes dans la langue française. Superposée au graphique à barres pour les fréquences de chaque lemme dans le corpus, cette figure inclut aussi la courbe de la loi de Zipf, qui dicte que de façon générale, que la fréquence d'utilisation d'un mot (ici un lemme) dans un texte volumineux est inversement proportionnelle à son rang (Zipf, 1945). Bien que l'accord entre les données « expérimentales » en bleu pour le corpus de référence et la courbe de la loi de Zipf en orange ne soit pas parfait, on constate tout de même que la tendance est bel et bien respectée.

Tableau 5.22 : Occurrences de lemmes les plus fréquents du roman « Le Rouge et le Noir », par classe grammaticale, avec nombre d'occurrences. Haut : verbes, adjectifs, noms communs et adverbes. Bas : déterminants, pronoms, prépositions, conjonctions et interjections

Verbes		Adjectifs		Noms communs		Adverbes	
Lemme	#	Lemme	#	Lemme	#	Lemme	#
être	4573	grand	466	monsieur	986	ne	2632
avoir	3436	petit	318	madame	606	pas	1115
dire	1274	jeune	295	homme	532	plus	1063
faire	1128	même	238	jour	399	bien	581
voir	617	bon	226	œil	283	tout	493
pouvoir	604	beau	200	marquis	282	si	410
aller	471	seul	196	heure	257	peu	302
trouver	400	premier	195	femme	255	fort	277
parler	382	autre	119	moment	253	jamais	267
vouloir	330	haut	94	air	249	encore	241

Déterminants		Pronoms		Prépositions		Conjonctions		Interjections	
Lemme	#	Lemme	#	Lemme	#	Lemme	#	Lemme	#
le	11595	il	6162	de	11169	et	3178	ah	90
un	5354	se	2284	à	4056	que	2279	eh	70
ce	3011	je	2091	en	1827	mais	791	hélas	27
son	2837	que	1639	dans	1400	comme	539	parbleu	8
mon	1399	lui	1587	pour	1339	si	473	bah	6
au	1277	qui	1499	avec	898	ou	331	adieu	5
de	1233	le	1444	par	803	quand	249	fi	4
quelque	366	vous	1139	sur	602	car	89	oh	2
tout	349	me	968	sans	384	donc	85	pardon	2
deux	297	ceci	770	après	314	puis	81	morbleu	2

Tableau 5.23 : Occurrences de lemmes les plus fréquents du roman de science-fiction, par classe grammaticale, avec nombre d'occurrences. Haut : verbes, adjectifs, noms communs et adverbes. Bas : déterminants, pronoms, prépositions, conjonctions et interjections

Verbes		Adjectifs		Noms communs		Adverbes	
Lemme	#	Lemme	#	Lemme	#	Lemme	#
être	3889	même	394	foi	407	ne	2311
avoir	3083	autre	393	colonel	357	pas	1742
faire	1090	grand	330	salle	295	plus	1224
pouvoir	555	jeune	317	contrôleur	285	bien	749
aller	530	seul	242	temps	259	aussi	360
savoir	347	bon	238	main	257	encore	342
dire	345	rouge	208	vaisseau	256	tout	279
devoir	335	terrien	192	œil	235	là	270
prendre	308	petit	169	terrien	234	toujours	263
falloir	287	premier	155	contrôle	219	non	262

Déterminants		Pronoms		Prépositions		Conjonctions		Interjections	
Lemme	#	Lemme	#	Lemme	#	Lemme	#	Lemme	#
le	12492	il	4105	de	9961	et	2553	ah	49
un	5430	se	3726	à	3494	que	2220	eh	33
son	2967	le	1267	en	2208	mais	1089	euh	29
ce	1970	je	1251	pour	1202	comme	412	merci	23
mon	1415	lui	1196	dans	1125	puis	349	bonjour	19
au	1241	on	1083	sur	1092	si	226	ouais	14
de	1219	qui	1060	avec	805	ou	177	hélas	13
deux	493	ceci	1051	par	771	donc	175	merde	9
quelque	443	que	967	vers	433	tandis	154	bravo	7
ton	297	ils	740	sans	321	quand	131	hello	7

Finalement, on remarque aussi que le lemme « le » se retrouve deux fois, dans chacune des deux figures. C'est qu'il y a deux lemmes distincts associés au mot « le », l'un étant un déterminant, l'autre étant un pronom personnel, et ceux-ci sont classés séparément, comme le veut la définition de lemme adoptée pour ce projet (Section 3.1.2). De la même façon, le mot « que » se répète deux fois à la Figure 5.27 (pronom et conjonction).

En observant les Figures 5.27 et 5.28, on note que la liste des 20 lemmes les plus courants est très semblable pour les deux corpus de référence. En effet, pas moins de 16 d'entre eux sont communs aux deux romans. On ne dispose ici que de deux exemples de textes, qui sont tous les deux des romans, mais on peut tout de même émettre l'hypothèse que les lemmes illustrés aux Figures 5.27 et 5.28 sont particulièrement courants dans la langue française, indépendamment du type de texte.

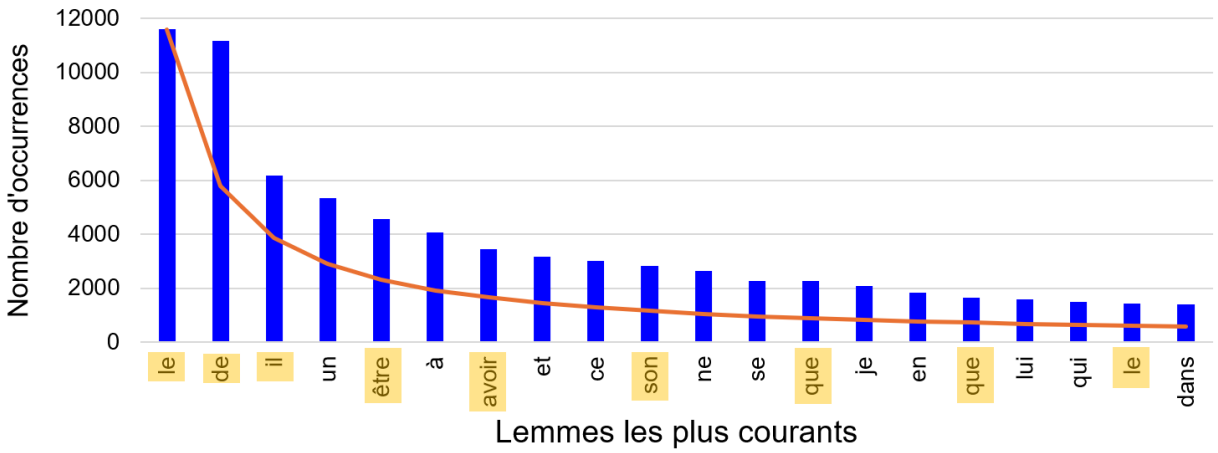


Figure 5.27 : Lemmes les plus fréquents du roman « Le Rouge et le Noir ». Les formes associées à des homographes sont surlignées en jaune. La courbe orange représente la loi de Zipf

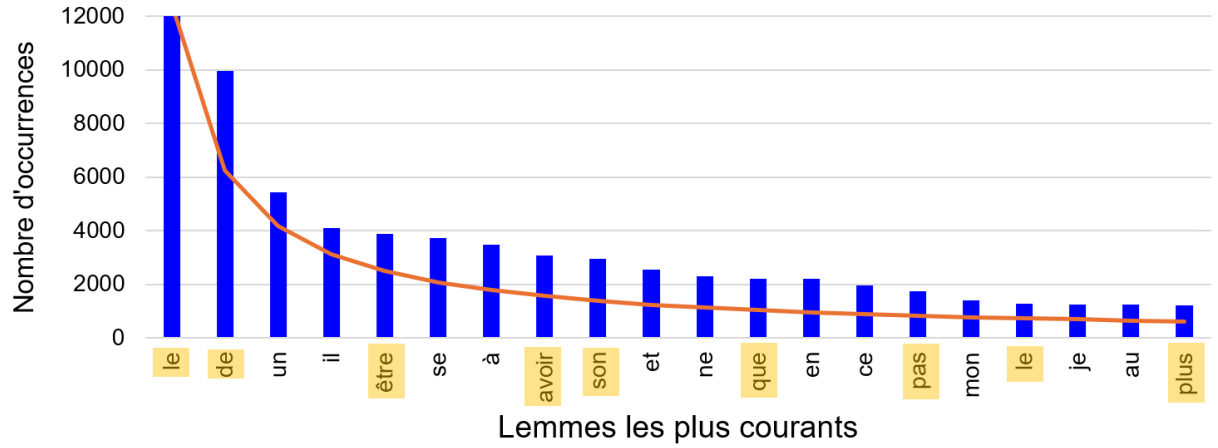


Figure 5.28 : Lemmes les plus fréquents du roman de science-fiction. Les formes associées à des homographes sont surlignées en jaune. La courbe orange représente la loi de Zipf

5.3.5.3. Proportions des classes grammaticales

A la Section 5.1.2, les proportions des neuf classes grammaticales pour le roman « Le Rouge et le Noir » avaient été présentées pour deux cas limites. Dans un premier cas, on ne considérait que les mots « non ambigus », donc en excluant les homographes. La deuxième approche tenait au contraire en compte toutes les possibilités de classes grammaticales pour chaque homographe. On se retrouvait ainsi avec un total inférieur à 100% dans le premier cas (puisqu'on ignorait tous les homographes), et un total supérieur à 100% dans le deuxième cas, puisqu'en grande majorité, les homographes contribuaient à plus d'une classe grammaticale. Ces deux cas limites permettaient donc d'estimer le minimum et le maximum possibles pour la proportion de chaque classe grammaticale. Ces valeurs minimales et maximales sont fournies au Tableau 5.4.

Mais une fois que la désambiguïsation des homographes a eu lieu, en utilisant les algorithmes décrits au Chapitre 4, il ne reste plus qu'une seule classe grammaticale possible pour chaque mot du corpus. Il devient donc possible d'obtenir une valeur unique (plutôt qu'une gamme de valeurs)

pour la proportion de chaque classe. Mais il faut tenir compte du fait que la désambiguïsation n'est pas parfaite comme on l'a vu aux Sections 5.3.3 et 5.3.4, où on fait état d'une performance tout de même enviable d'environ 95%. Comme première étape, on compare donc les proportions de classes grammaticales obtenues à la suite d'une désambiguïsation manuelle avec celles obtenues par l'algorithme de désambiguïsation. Comme seuls les 11 024 premiers homographes du roman « Le Rouge et le Noir » ont été désambiguïsés manuellement, on s'attardera uniquement à ces 11 024 homographes dans les deux cas (Tableau 5.24). On constate que les deux ensembles de valeurs sont très semblables. En effet, le plus grand écart n'est que de 0.7%, pour les déterminants et les prépositions, l'écart moyen étant autrement que de 0.3%. On constate donc que l'algorithme de désambiguïsation a permis de correctement évaluer la proportion des classes grammaticales dans le corpus.

Sur la base de ce résultat, on peut donc se fier à l'algorithme, et ainsi l'appliquer au corpus en entier, plutôt que de se limiter aux 11 024 premiers homographes tel qu'on l'a fait pour le Tableau 5.24. Le Tableau 5.25 illustre où se situent les proportions de classes grammaticales, en rapport aux deux cas limites du Tableau 5.4 présenté précédemment. On constate, tel que prévu, que les proportions après désambiguïsation des homographes se situent effectivement entre les valeurs minimales et maximales obtenues avant désambiguïsation, dans les 9 cas.

Tableau 5.24: Proportions des neuf classes grammaticales selon les fréquences dans le roman « Le Rouge et le Noir » pour les homographes uniquement, après désambiguïsation manuelle et par l'algorithme (11 024 premiers homographes)

Classe grammaticale	Désambiguïsation manuelle	Désambiguïsation par l'algorithme
Verbes	10%	10%
Adjectifs	11%	11%
Noms	14%	15%
Adverbes	7.3%	7.1%
Déterminants	25%	24%
Pronoms	7.4%	7.8%
Prépositions	20%	21%
Conjonctions	5.3%	4.9%
Interjections	0.045%	0.045%
TOTAL	100%	100%

Tableau 5.25: Proportions des neuf classes grammaticales selon les fréquences dans le roman « Le Rouge et le Noir » avant désambiguïsation (en incluant et en excluant les homographes) et après désambiguïsation

Classe grammaticale	AVANT désambiguïsation En excluant les homographes	**APRÈS désambiguïsation**	AVANT désambiguïsation En assignant toutes les classes possibles de chaque homographe
Verbes	13%	**17%**	23%
Adjectifs	3.5%	**6.7%**	11%
Noms	14%	**22%**	25%
Adverbes	4.6%	**6.9%**	7.7%
Déterminants	8.8%	**16%**	24%
Pronoms	11%	**13%**	22%
Prépositions	7.0%	**13%**	13%
Conjonctions	3.1%	**4.6%**	6.2%
Interjections	0.10%	**0.12%**	0.13%
TOTAL	65%	100%	131%

5.3.5.4. Proportions des personnes de verbe après désambiguïsation

En ce qui concerne les temps et personnes de verbes, la désambiguïsation n'a pas pu ici jouer un très grand rôle, puisque limitée aux homographes de classes grammaticales différentes. En effet, la difficulté inhérente à la détermination des temps et des personnes de verbes relève principalement de la présence de multiples formes verbales identiques pour un verbe donné. On a déjà donné en exemple la forme verbale « aime », qu'on retrouve en cinq combinaisons de temps et personnes pour le verbe « aimer ». Une désambiguïsation appropriée des temps et personnes de verbes demanderait une analyse spécifiquement dédiée à ce problème.

Les résultats après désambiguïsation sont tout de même plus précis que ceux illustrés au Tableau 5.5, car la désambiguïsation a permis d'éliminer les homographes pouvant en théorie être des formes verbales, mais qui, dans le contexte du corpus, appartenaient à une autre classe grammaticale. Par exemple, dans la phrase « Je lui ai fait la grande demande », le mot demande aurait été inclus dans l'analyse des formes verbales au Tableau 5.5, car on ne savait encore à ce point si « demande » était un verbe ou un nom commun. Mais après désambiguïsation (en la supposant efficace), on peut exclure le mot « demande » de cette phrase de notre analyse, puisqu'il s'agit ici d'un nom commun.

Mais comme l'ambiguïté demeure, de la même façon qu'on l'avait fait au Tableau 5.5., on fournit au Tableau 5.26 les proportions de personnes de verbe pour deux scénarios distincts : en incluant les formes ambiguës, puis en les excluant. En comparant avec les valeurs obtenues précédemment au Tableau 5.5 (avant désambiguïsation), on constate que les gammes de valeurs possibles ont été réduites substantiellement. Par exemple, la gamme de valeurs de proportion de la première personne du singulier est passée de 2.3%-30% (Tableau 5.5) à 3.1%-16% (Tableau 5.26). On observe le même comportement pour les autres personnes.

Tableau 5.26: Proportions des personnes de verbe selon les fréquences dans le roman « Le Rouge et le Noir » après désambiguïsation (en incluant et en excluant les cas ambigus)

Personnes de verbe	En excluant les cas ambigus (possibilité de plus d'une personne de verbe)	En assignant toutes les personnes possibles à chaque forme verbale
1ère du singulier	3.1%	16%
2e du singulier	0.37%	14%
3e du singulier	63%	78%
1ère du pluriel	0.25%	0.72%
2e du pluriel	1.8%	4.3%
3e du pluriel	5.6%	7.7%
TOTAL	75%	120%

5.3.5.5. Proportions des temps de verbe après désambiguïsation

Tout comme on l'a fait à la section précédente pour les personnes de verbe, on peut maintenant présenter les proportions de temps de verbe après désambiguïsation. Encore là, de l'ambiguïté demeure, comme l'identification des temps de verbe est hors de la portée du présent projet. Tout de même, la désambiguïsation a permis de limiter l'analyse aux seuls mots qui ont été classifiés comme formes verbales, éliminant ainsi des homographes de type « verbe-nom » par exemple, qui étaient inclus dans l'analyse précédente, qu'ils soient avérés ou non être des formes verbales. Les gammes de valeurs de proportions de temps de verbe affichées au Tableau 5.27 sont donc plus précises que celles affichées au Tableau 5.6 avant désambiguïsation.

Tableau 5.27: Proportions des temps de verbe selon les fréquences dans le roman « Le Rouge et le Noir » après désambiguïsation (en incluant et en excluant les cas ambigus)

Temps de verbe	En excluant les cas ambigus (possibilité autre classe, ou plus d'une personne de verbe)	En assignant toutes les personnes possibles à chaque forme verbale
Infinitif	20%	20%
Présent	12%	27%
Imparfait	20%	21%
Passé simple	14%	18%
Futur simple	3.1%	3.2%
Subjonctif présent	0.31%	7.3%
Subjonctif imparfait	1.6%	1.6%
Impératif	0.017%	9.0%
Conditionnel	1.5%	2.1%
Participe présent	3.1%	3.1%
Passé composé	5.8%	8.2%
TOTAL	81%	121%

On note que les gammes de valeurs ont été substantiellement réduites en comparaison avec celles du Tableau 5.6 obtenues avant désambiguïsation. En particulier, pour les formes infinitives, on obtient le même résultat en incluant ou excluant les formes ambiguës, ce qui implique qu'aucune forme infinitive ne demeure ambiguë. Cela s'explique par le fait que les formes infinitives sont toujours distinctes des formes conjuguées, et que l'ambiguïté du Tableau 5.6 était due au fait que plusieurs verbes à l'infinitif peuvent aussi appartenir à d'autres classes grammaticales. On a par exemple, les homographes « dîner », « pouvoir » et « devoir ». La désambiguïsation a pu éliminer de telles ambiguïtés.

On retrouve beaucoup moins de cas non ambigus pour les autres temps, ce qui témoigne du fait qu'en français, la plupart des formes verbales sont ambiguës. L'exemple typique est la forme conjuguée « aime » qu'on retrouve en tout dans cinq cas (1ère et 3e personnes du singulier du présent et du subjonctif, ainsi qu'à la 2e personne du singulier de l'impératif).

5.4. Analyse des cooccurrences

L'Étape 1 du projet comprend aussi la détermination des cooccurrences pour les lemmes de chaque verbe, adjectif, nom commun ou adverbe du corpus de référence. Ces cooccurrences ne sont utiles ni pour la lemmatisation, ni pour la désambiguïsation du corpus de référence. En revanche, cette information est utile à l'Étape 2 pour la génération de textes aléatoires automatiquement lemmatisés. En effet, à cette étape, on influence la sélection des mots au hasard en fonction de ces cooccurrences.

Tel que précisé à la Section 4.10.1, pour chaque lemme appartenant à l'une des quatre classes grammaticales citées plus haut, on a associé quatre tables de hachage, encore là associées aux verbes, adjectifs, noms communs et adverbes. Ces tables de hachage renferment les cooccurrences associées à chaque lemme, sous chacune de ces classes. Selon l'analyse des classes grammaticales des deux corpus de référence, on retrouve 6076 lemmes distincts appartenant à ces quatre classes dans le roman « Le Rouge et le Noir », et 7623 lemmes distincts dans le roman de science-fiction.

Et puisqu'à chacun de ces lemmes on associe quatre tables de hachage, on se retrouve ainsi avec environ 25 000 à 30 000 tables de hachage dans lesquelles on emmagasine les données pour les cooccurrences. Il n'est donc pas possible d'illustrer tous ces résultats dans ce mémoire. Au Tableau 5.28, on fournit le nombre de cooccurrences distinctes, par classe grammaticale, pour les 10 lemmes de noms communs les plus courants identifiés précédemment au Tableau 5.22, pour le roman « Le Rouge et le Noir ». On constate, règle générale, que c'est à la colonne des cooccurrences de noms qu'on retrouve le plus de cooccurrences, suivi des verbes, puis non loin derrière des adjectifs. On constate aussi, sans grande surprise, que plus un lemme est courant, plus on lui associe de cooccurrences. À l'inverse, un lemme peu courant, par exemple le nom « couvent » qu'on ne retrouve que 7 fois dans le roman, n'est associé qu'à 76 cooccurrences en tout (dont les noms « cœur » et « esprit », et l'adjectif « sacré »).

À titre d'exemple, le Tableau 5.29 fournit les cooccurrences les plus courantes du lemme « cœur », en fonction des classes grammaticales. On note certains mots ayant un lien sémantique avec le mot « cœur ». On retrouve par exemple les adjectifs « grand », « sacré » et « bon », ainsi que les noms « amour » et « foi », qu'on peut facilement associer à « cœur ». Mais toujours est-il que ces mots sémantiquement associés à « cœur » se retrouvent quelque peu noyés parmi toutes les autres cooccurrences. En effet, on retrouve principalement parmi ces cooccurrences les lemmes les plus courants du corpus, sans lien direct avec le mot « cœur ».

Tableau 5.28: Nombre de cooccurrences associées aux 10 lemmes de noms communs les plus courants dans le roman « Le Rouge et le Noir », selon les classes grammaticales

Lemme (nom)	Nombre de cooccurrences			
	Verbes	Adjectifs	Noms	Adverbes
monsieur	559	535	1135	137
homme	418	414	835	122
madame	417	375	780	113
jour	348	324	724	87
œil	311	307	600	87
femme	273	256	491	85
moment	288	234	499	74
air	237	291	470	83
marquis	261	231	473	70
heure	256	205	475	70

Tableau 5.29: Liste des cooccurrences associées au lemme « cœur » dans le roman « Le Rouge et le Noir », selon les classes grammaticales. Nombre d'occurrences fourni dans chaque cas

Verbes		Adjectifs		Noms		Adverbes	
Lemme	Nombre	Lemme	Nombre	Lemme	Nombre	Lemme	Nombre
être	107	grand	12	homme	29	ne	64
avoir	94	sacré	12	madame	19	pas	33
faire	30	petit	10	monsieur	13	plus	29
dire	26	bon	10	amour	10	tout	18
savoir	14	jeune	10	mot	10	bien	16
pouvoir	12	seul	9	jour	8	toute	12
parler	11	même	6	moment	7	si	11
voir	11	faible	5	esprit	7	peu	10
comprendre	10	intime	4	foi	6	même	9
passer	9	honnête	4	œil	6	jamais	7

Le Tableau 5.30 fournit quant à lui la liste de cooccurrences associées au lemme « porte », toujours dans le roman « Le Rouge et le Noir », à titre d'exemple supplémentaire. La liste de lemmes de verbes (première colonne) est très semblable à ce qu'on avait retrouvé pour le lemme « cœur ». En effet, la liste est dominée par les verbes les plus courants du corpus (« être », « avoir », « faire », « dire », etc.). Mais on retrouve aussi au Tableau 5.30 les verbes « ouvrir » et « fermer », qui eux, ont un lien sémantique fort avec le lemme « porte ». De la même façon, on retrouve au sein de la liste d'adjectifs, les lemmes « ouvert » et « fermé », équivalents aux verbes « ouvrir » et « fermer ». Le reste des adjectifs cooccurrents se compose encore essentiellement des adjectifs génériques les plus courants du corpus. Pour ce qui est des noms, on retrouve dans la liste « clef » et « chambre », ayant eux aussi un fort lien sémantique avec le lemme « porte ». On observe donc un certain nombre de cooccurrences au Tableau 5.30 offrant un lien sémantique avec le lemme auquel elles se rapportent. Mais une fois de plus, ces cooccurrences pertinentes se retrouvent noyées parmi celles plus génériques communes à pratiquement tous les lemmes du corpus.

Tableau 5.30 : Liste des cooccurrences associées au lemme « porte » dans le roman « Le Rouge et le Noir », selon les classes grammaticales. Nombre d'occurrences fourni dans chaque cas

Verbes		Adjectifs		Noms		Adverbes	
Lemme	Nombre	Lemme	Nombre	Lemme	Nombre	Lemme	Nombre
être	68	petit	11	heure	14	ne	22
avoir	21	grand	9	chambre	12	tout	16
faire	15	jeune	7	clef	11	près	9
ouvrir	15	autre	4	monsieur	8	pas	8
pouvoir	11	bon	4	coup	7	plus	8
entrer	9	ouvert	3	matin	7	peu	6
voir	8	fermé	3	prison	6	presque	5
étayer	6	haut	3	homme	6	bien	5
dire	6	possible	3	bibliothèque	6	fort	5
fermer	6	doré	3	lendemain	5	enfin	4

On ne peut donc pas s'attendre à ce que l'application de cooccurrences pour générer des phrases aléatoires à l'Étape 2 mène à des combinaisons de mots intimement associés, comme on l'aurait espéré, à moins de « corriger » les fréquences pour diminuer l'importance des cooccurrences très courantes dans le corpus. Une telle correction a en effet été apportée, en « normalisant » les fréquences d'apparition des cooccurrences pour un lemme en particulier en fonction des fréquences de ces mots dans le corpus en entier. On s'est ainsi retrouvé à favoriser les cooccurrences spécifiques au lemme sous étude.

Les résultats n'ont été fournis ici que pour deux lemmes (« cœur » et « porte »), mais des tableaux équivalents ont été générés pour tous les lemmes du corpus parmi les classes grammaticales verbes, adjectifs, noms communs et adverbes. Et ce sont ces tableaux qui ont influencé la sélection des mots de ces mêmes classes à l'Étape 2 du projet, pour la génération de phrases aléatoires.

5.4.1. Informations fournies en sortie de la lemmatisation pour l'Étape 2

Tel que mentionné à la Section 3.5.1, on doit fournir en sortie de l'Étape 1 une liste des lemmes accompagnés de leurs fréquences d'apparition dans le corpus, pour les verbes, adjectifs, noms communs et adverbes. Cette information est emmagasinée sous forme de tables de hachage. Ce sont les statistiques *après désambiguïsation* qui sont utilisées par défaut, pour s'assurer d'une plus grande fidélité au corpus de référence. Il n'est pas pertinent ici de reproduire cette information. Toutes les tables de hachage des cooccurrences générées à l'Étape 1, discutées à la Section 5.4, sont aussi fournies pour contribuer à la sélection des mots au hasard à l'Étape 2.

On doit aussi fournir les fréquences des personnes et des temps de verbes retrouvées dans le corpus de référence. Cette information a été fournie aux Tableaux 5.26 et 5.27, sous forme de limites inférieures et supérieures dans chaque cas. Pour générer les textes aléatoires, il a donc fallu choisir les fréquences en fonction du cas « incluant les ambigus » ou du cas les excluant. En effet, tel que mentionné à la Section 5.3.5.4, aucun effort n'a été déployé au cours de ce projet pour désambiguïser entre elles les formes verbales, en ce qui concerne les temps et les personnes. Par défaut, l'approche excluant les ambigus a été sélectionnée ici, considérée comme étant plus représentative.

Toutefois, tel que mentionné à la Section 3.5.3.1, l'algorithme offre d'imposer un seuil minimum à la fois pour les temps et les personnes de verbes. L'imposition de tels seuils fait en sorte que les

phrases générées aléatoirement s'éloignent quelque peu du style adopté dans le corpus de référence. Cependant, il faut préciser que le but de ces seuils est de s'assurer, optionnellement, que tous les temps de verbes et que chacune des six personnes soient représentés parmi les phrases générées aléatoirement. En particulier, l'imposition de tels seuils assure que les temps composés, non détectés automatiquement, puissent se retrouver dans les phrases générées aléatoirement. En effet, il est pertinent d'inclure toutes les possibilités de temps et personnes de verbes dans une certaine proportion, afin de s'assurer que les outils de lemmatisation puissent y être confrontés. L'évaluation de lemmatiseurs existants sera discutée au Chapitre 6.

Pour les personnes de verbes, on a imposé un seuil minimal de 10% pour chaque personne, pour la création de phrases aléatoires. Le Tableau 5.31 fournit donc, pour les deux corpus utilisés, le pourcentage détecté de chacune des personnes lors de la lemmatisation (basé sur les cas non ambigus) ainsi que le pourcentage modifié pour obtenir au moins le seuil minimal dans chacun des cas. On note que pour le roman « Le Rouge et le Noir », seule la 3^{e} personne du singulier dépassait ce seuil. Les autres personnes se sont donc vu imposer une proportion de 10%, laissant ensuite le 50% restant pour la 3^{e} personne du singulier. Dans le cas du roman de science-fiction, la 3^{e} personne du pluriel dépassait aussi la valeur seuil de 10%, si bien que les proportions modifiées ont dû en tenir compte.

Pour la génération des phrases aléatoires, un seuil minimal de 2% a été imposé pour tous les temps. Le Tableau 5.32 fournit donc les fréquences détectées puis modifiées de temps de verbes, incorporant ces seuils minimaux, pour les deux corpus. On note toutefois que pour cet exercice, on exclut l'infinitif, le participe présent et le participe passé. Ces temps sont exclus car on ne démarrera pas le groupe du verbe à l'Étape 2 avec un verbe à l'un de ces temps. Tel que mentionné plus haut, aucun temps composé n'est détecté automatiquement par l'algorithme de lemmatisation, ce qui explique les valeurs nulles pour ces temps. Il est à noter que la somme des pourcentages de chaque colonne donne un total de 100%.

Tableau 5.31 : Fréquences détectées des personnes de verbe dans les deux corpus de référence, puis modifiées pour assurer un seuil minimum de 10% pour chacune d'elles au sein des phrases aléatoires

	Le Rouge et le Noir		Science-fiction	
	Détectées	Modifiées	Détectées	Modifiées
1ère singulier	4.2%	10%	3.6%	10%
2e singulier	0.49%	10%	2.86%	10%
3e singulier	85%	50%	79%	49%
1ère pluriel	0.34%	10%	0.56%	10%
2e pluriel	2.4%	10%	2.1%	10%
3e pluriel	7.6%	10%	11.9%	11%

Tableau 5.32 : Fréquences détectées des temps de verbe dans les deux corpus de référence, puis modifiées pour assurer un seuil minimum de 2% pour chacun d'eux au sein des phrases aléatoires

	Le Rouge et le Noir		Science-fiction	
	Détectées	Modifiées	Détectées	Modifiées
Infinitif	25%	0%	30%	0.0%
Présent	14%	19%	40%	61%
Imparfait	25%	32%	5.1%	7.7%
Passé simple	17%	22%	0.34%	2.0%
Futur simple	3.9%	5.0%	3.1%	4.7%
Subjonctif présent	0.39%	2.0%	1.2%	2.0%
Subjonctif imparfait	2.0%	2.0%	0.026%	2.0%
Impératif	0.021%	2.0%	0.020%	2.0%
Conditionnel	1.8%	2.0%	2.7%	4.2%
Participe présent	3.8%	0.0%	7.3%	0.0%
Participe passé	7.1%	0.0%	10%	0.0%
Passé composé	0.0%	2.0%	0.0%	2.0%
Plus-que-parfait	0.0%	2.0%	0.0%	2.0%
Passé antérieur	0.0%	2.0%	0.0%	2.0%
Futur antérieur	0.0%	2.0%	0.0%	2.0%
Subjonctif passé	0.0%	2.0%	0.0%	2.0%
Subjonctif plus-que-parfait	0.0%	2.0%	0.0%	2.0%
Conditionnel passé	0.0%	2.0%	0.0%	2.0%

5.5. Vitesse d'exécution : lemmatisation, désambiguïsation, cooccurrences

La rapidité d'exécution des différents algorithmes n'est pas l'objectif principal de ce projet. Le but principal est plutôt de fournir des textes générés aléatoirement automatiquement lemmatisés, pour éventuellement s'en servir d'étalon pour l'évaluation d'outils de lemmatisation existants. Que toutes les étapes menant à ces textes aléatoires soient exécutées en quelques secondes ou en quelques minutes n'est pas critique. En effet, pour un corpus de référence donné, c'est là une opération qu'un utilisateur n'exécute potentiellement qu'au plus quelques fois, en variant quelques paramètres. Il ne s'agit pas d'une opération devant se répéter des centaines, voire des milliers de fois.

Toujours est-il qu'une vitesse d'exécution rapide comporte son lot d'avantages, incluant la possibilité de varier les paramètres d'entrée comme la proportion de temps de verbe, de phrases à la forme négative, etc. et de rapidement constater l'effet de ces changements sur les résultats obtenus. Une exécution rapide pourrait aussi éventuellement permettre une adaptation de ces algorithmes pour les intégrer à un outil de traitement de texte, pour effectuer cette opération « en temps réel », bien au-delà du projet actuel. Finalement, une exécution rapide a permis un développement rapide et efficace des algorithmes, ce qui comprend évidemment toutes les opérations de débogage, inévitables dans un projet de cette envergure. Le but de cette section est donc de donner une idée de la vitesse d'exécution de l'Étape 1 de ce projet, qui comprend la lemmatisation de base, la désambiguïsation des homographes, et la génération de cooccurrences.

Le temps d'exécution dépend d'un facteur principal qui est la taille du corpus de référence. Ce facteur est donc varié pour en étudier l'influence sur le temps d'exécution, ce qui permet de déterminer l'ordre algorithmique de ce processus de lemmatisation. Il faut toutefois se rappeler que la désambiguïsation des homographes s'effectue en deux étapes, soit l'entrainement du modèle d'apprentissage machine, puis l'application du modèle. Des mesures de rapidité distinctes seront donc effectuées pour ces deux étapes. Pour l'entrainement, on a utilisé l'approche manuelle pour évaluer la rapidité d'exécution. Pour l'application, l'approche gauche-droite de l'analyse de la phrase a été sélectionnée.

Le temps d'exécution dépend forcément aussi de la vitesse et de l'efficacité de l'ordinateur utilisé. Mais le rôle de la machine utilisée n'étant pas pertinent ici, cette variable sera ignorée, en mettant l'emphase sur les temps d'exécution *relatifs*.

Les corpus de référence utilisés pour ce projet, tels que décrits à la Section 3.4 comprennent environ 180 000 mots chacun et plus d'un million de caractères, en comptant les espaces. Pour étudier la rapidité d'exécution, on a utilisé le roman « Le Rouge et le Noir », et créé quelques versions tronquées de ce corpus de référence (facteurs de 25 000). Le plus grand ensemble utilisé comprend 175 000 caractères, car l'analyse présente s'est limitée à la portion du corpus lemmatisée manuellement, afin que les processus d'entrainement et d'évaluation s'effectuent sur l'ensemble du texte tronqué.

Le temps d'exécution a été mesuré en tirant profit de la fonction « *System.currentTimeMillis()* » de Java. Cette fonction fournit le temps en millisecondes depuis une date de référence en particulier. Il nous suffit donc d'obtenir la valeur de cette fonction au début de l'exécution et à la fin, et ensuite de soustraire ces deux valeurs pour obtenir le temps total d'exécution de l'Étape 1. Ces valeurs sont fournies au Tableau 5.33, transformées en secondes. Ces mêmes données sont exprimées graphiquement à la Figure 5.29. On constate que l'ordre algorithmique dans ces trois cas est approximativement linéaire. C'est donc dire que les durées d'exécution augmentent linéairement en fonction de la taille du corpus analysé.

Tableau 5.33 : Durée d'exécution pour l'Étape 1 de ce projet, en fonction du nombre de caractères inclus dans le corpus de référence tronqué, et selon l'opération effectuée (entrainement ou application de l'apprentissage machine). L'appareil utilisé est un ordinateur personnel Acer Aspire X3470 muni d'un processeur AMD A6-3620 APU de 2.20 GHz

	Durée d'exécution (secondes)		
# caractères	Entrainement	Application	
		Gauche-droite	Phrase complète
25000	21	14	56
50000	28	14	99
75000	37	16	139
100000	43	19	175
125000	52	21	206
150000	61	23	233
175000	64	34	267

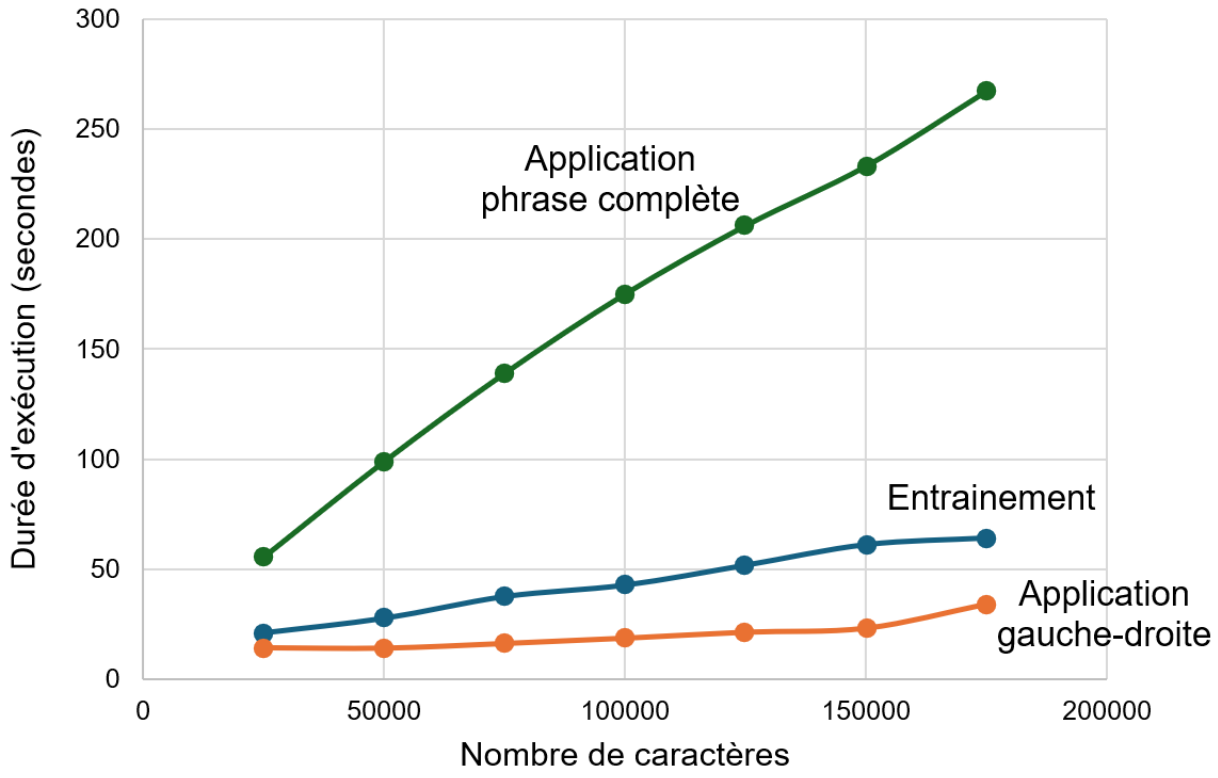


Figure 5.29 : Durée d'exécution pour l'Étape 1 de ce projet, en fonction du nombre de caractères inclus dans le corpus de référence tronqué, et selon l'opération effectuée (entrainement ou application de l'apprentissage machine). Données tirées du Tableau 5.33

Les durées d'exécution du Tableau 5.33 considèrent un grand nombre d'étapes effectuées pour l'Étape 1. À titre de référence, le Tableau 5.34 liste les durées d'exécution des principaux processus de l'Étape 1 pour l'entrainement et l'application de l'apprentissage machine. Les mêmes résultats sont fournis sous forme graphique à la Figure 5.30. Les données utilisées ont été obtenues en utilisant la partie lemmatisée manuellement du roman « Le Rouge et le Noir ».

En premier lieu, on note au Tableau 5.34 que la durée totale de l'entrainement est plus grande que l'évaluation, ce qui est en accord avec les données du Tableau 5.33 cité précédemment, dans le cas de l'application selon l'approche « gauche-droite ». On constate ensuite que les étapes les plus longues sont l'entrainement et la désambiguïsation, deux processus centraux à l'algorithme global développé. Au-delà de la compilation du code Java, une autre étape relativement importante est celle de la création des centaines de milliers de formes verbales, issues des milliers de verbes du Bescherelle et des dizaines de combinaisons de temps et de personnes possibles pour chacun d'eux. Cette étape ne prend tout de même que moins de trois secondes à s'exécuter (indépendamment de la taille du corpus à analyser). On constate aussi que plusieurs étapes s'effectuent très rapidement, comme la lemmatisation de base, l'inventaire des homographes ainsi que l'analyse des cooccurrences. On observe aussi au Tableau 5.34 que les étapes communes aux deux grandes opérations (entrainement et application) s'effectuent en environ le même temps.

Tableau 5.34 : Durées d'exécution pour différents processus de l'Étape 1, obtenus en utilisant la partie lemmatisée du roman « Le Rouge et le Noir » pour l'entrainement et l'application du modèle d'apprentissage machine

	Temps d'exécution (secondes)	
Processus	Entrainement	Application
Compilation du code Java	11	11
Banque de mots: formes verbales	2.8	2.8
Banque de mots: autres classes grammaticales	0.80	0.89
Lire et nettoyer le corpus	0.54	0.53
Lemmatisation de base	0.40	0.37
Inventaire des homographes	0.26	0.18
Entrainement	56	
Désambiguïsation		14
Compiler résultats et enregistrer	0.90	0.93
Analyse des cooccurrences	0.26	0.18
Total	73	31

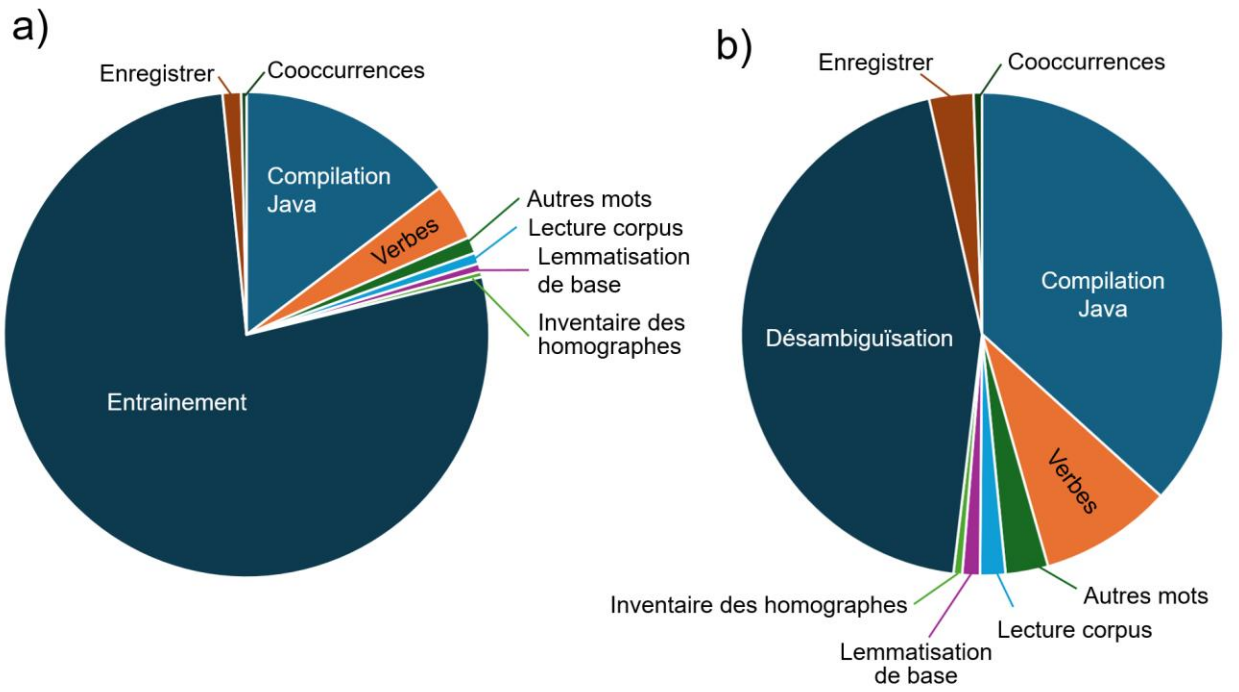


Figure 5.30 : Durées d'exécution relatives pour différents processus de l'Étape 1, obtenus en utilisant la partie lemmatisée du roman « Le Rouge et le Noir » pour l'entrainement (a) et l'application (b) du modèle d'apprentissage machine. Données tirées du Tableau 5.34

5.6. Génération de phrases aléatoires automatiquement lemmatisées

L'Étape 2 représente l'objectif ultime de ce projet qui consiste à générer des phrases aléatoires automatiquement lemmatisées. L'algorithme développé pour l'Étape 2 reçoit en entrée des banques de données et paramètres fournis par les résultats de l'Étape 1, discutée à la section précédente (Section 5.4.1). En sortie de l'Étape 2, on fournit des textes qui pourront être directement utilisés pour l'évaluation d'outils de lemmatisation existants, au Chapitre 6. Les tableaux fournis et les statistiques globales pourront alors être comparés avec l'information similaire fournie par ces outils existants.

Mais en plus des statistiques issues du corpus de référence sélectionné, certains paramètres plus ou moins arbitraires influencent la création des phrases. Ces paramètres sont discutés à la section suivante.

5.6.1. Liste et sélection des différents paramètres

A la Section 5.4.1, on mentionne toutes les informations générées à l'Étape 1, requises pour l'exécution de l'Étape 2. On y retrouve entre autres les listes et fréquences de verbes, adjectifs, noms et adverbes utilisés, des proportions de temps et de personnes de verbes, ainsi que des tableaux de cooccurrences. Mais au-delà ces informations, d'autres paramètres doivent aussi être spécifiés pour générer les phrases. Ces paramètres, non inspirés de l'analyse du corpus de référence, sont donc déterminés arbitrairement. On aurait pu dans certains cas incorporer une analyse du corpus de référence pour déterminer ces paramètres de façon automatique, mais tel n'a pas été le cas ici. On retrouve la liste exhaustive de ces paramètres au Tableau 5.35.

Le hasard intervient aussi dans d'autres cas, comme par exemple pour déterminer le genre et le nombre des noms, adjectifs et participes passés, la personne pour l'impératif, le choix de l'auxiliaire quand un participe passé peut être accompagné autant par l'auxiliaire « avoir » que « être », le choix d'une préposition suivant le verbe quand plusieurs options sont possibles, et la position de l'adverbe quand plusieurs options sont possibles. Mais dans tous ces cas, des probabilités égales ont été imposées pour toutes les options possibles. Aucun paramètre additionnel n'a donc été introduit pour ces cas précis.

Au sein des groupes du sujet et du nom, quand l'option d'un pronom est déterminée, il faut ensuite spécifier le type de pronom à être utilisé. Le Tableau 3.28 au Chapitre 3 fournit la liste complète de toutes les options considérées pour ce projet, ainsi que certaines contraintes concernant leur utilisation. Le Tableau 5.36 fournit quant à lui les « poids » arbitraires associés à chacune de ces options, pour la sélection au hasard du type de pronom à utiliser. Ces poids déterminent l'importance relative de chaque type de pronom pour leur sélection aléatoire. Si le même poids était accordé à tous les types, leurs probabilités respectives seraient donc égales. Plus un poids est élevé, plus la probabilité de sélectionner le pronom en question est élevée. Comme on le voit au Tableau 5.36, on accorde donc arbitrairement une plus grande importance aux pronoms personnels-sujet et aux pronoms possessifs, en comparaison avec les autres types de pronoms.

De la même façon, dans le cas où le groupe du sujet ou du nom est basé sur les noms communs, il faut spécifier le type de déterminant à inclure. Le Tableau 3.30 au Chapitre 3 fournit la liste complète de toutes les options considérées pour ce projet, ainsi que certaines contraintes concernant leur utilisation. Le Tableau 5.37 fournit quant à lui les « poids » associés à chacune de ces options, pour la sélection au hasard du type de déterminant à utiliser. Ces poids déterminent l'importance relative de chaque type de déterminant.

Tableau 5.35 : Liste de paramètres, avec leurs valeurs par défaut, influençant la création de phrases aléatoires. Ces valeurs sont des probabilités. Par exemple, 0.35 correspond à une probabilité de 35%

Partie de la phrase	Paramètre	Valeur par défaut	Explication
Groupe verbe	probModal	0.3	Probabilité qu'un verbe apparaisse sous la forme modale, donc incluant un verbe modal suivi d'un infinitif (je *dois* aller, je *veux* marcher, etc.)
Groupe verbe	probNegatif	0.3	Probabilité que le verbe soit à la forme négative (je *ne* veux *pas* manger)
Groupe verbe	probAdverbe	0.4	Probabilité que le verbe du groupe du verbe soit accompagné d'un adverbe de caractérisation (je cours *vite*)
Groupe verbe	probIntensite	0.3	Probabilité que si un adverbe de caractérisation est inclus dans le groupe du verbe, celui-ci soit à son tour accompagné d'un adverbe d'intensité (je cours *très* vite)
Groupe sujet/nom	probSubjNeg	0.3	Probabilité que la forme subjonctive du type « il faut que » incluse au groupe du nom prenne la forme négative (il *ne* faut *pas* qu'il vienne)
Groupe sujet/nom	probNom	0.9	Probabilité que le groupe du sujet ou du nom soit composé de noms (le *chien*, ces *camions*) plutôt que de pronoms (*il*, *ceux-ci*)
Groupe sujet/nom	prob2noms	0.5	Probabilité que si le groupe sujet est basé sur les noms, qu'il contienne deux noms plutôt qu'un (*les chiens et les chats* se détestent)
Groupe sujet/nom	probAdjectif	0.6	Probabilité qu'un nom commun soit accompagné d'un adjectif (le camion *rouge*)
Groupe sujet/nom	probAdverbe	0.4	Probabilité que si un adjectif est inclus, il soit accompagné d'un adverbe d'intensité (la *très* grosse voiture)
Groupe sujet et Groupe complément	probCompDu	0.3	Probabilité d'inclure un complément du nom de type « du » (la voiture *du voisin* »)
Groupe sujet et Groupe complément	probCompQue	0.5	Probabilité d'inclure un complément du nom de type « que » (la femme *que je connais*)

Comme on le constate au Tableau 5.37, on accorde ici arbitrairement une plus grande importance aux articles ainsi qu'aux déterminants démonstratifs et possessifs, en comparaison avec les autres types de déterminants.

Quand il est question d'articles définis et indéfinis, ainsi que de déterminants démonstratifs, le choix du déterminant dépend ensuite du genre (féminin ou masculin) et du nombre (singulier ou pluriel) du nom que le déterminant accompagne. Ce choix ne s'effectue donc pas au hasard. Pour ce qui est des déterminants possessifs, en plus du genre et du nombre, il faut se soucier de la « personne » a qui « appartient » le nom, soit l'une des trois personnes, au pluriel et au singulier. On fait donc à ce moment intervenir le hasard pour choisir l'une de ces six personnes. Mais aucun paramètre ne doit être spécifié, car un poids égal est attribué à chacune d'elles.

On note finalement que certains déterminants du Tableau 5.37 font intervenir deux mots, comme « un tel » ou « un quelconque ». Le cas « n'importe quel » fait même intervenir trois mots.

Tableau 5.36 : Liste des poids associés arbitrairement à chaque type de pronom, influençant la sélection aléatoire de ceux-ci

Type de pronom	Poids relatif
Personnel-sujet (je, tu, il…)	10
Possessif (le mien, le tien…)	10
Tous / toutes	5
Celui-ci, celui-là, celle-ci, celle-là	5
Personne	5
Quelqu'un, quelques-uns, quelques-unes	5
Rien	5
Quiconque	5
N'importe lequel / laquelle / lesquels / lesquelles / qui	5

Tableau 5.37 : Liste des poids associés arbitrairement à chaque type de déterminant, influençant la sélection aléatoire de ceux-ci

Type de déterminant	Liste	Poids relatif
Article défini	le, la, les	20
Article indéfini	un, une, des	20
Démonstratif	ce, cette, ces	10
Possessif	mon, ton, son, ma, ta, sa, notre, votre, leur, mes, tes, ses, nos, vos, leurs	10
« chaque »		1
« divers »		1
« quelques »		1
« nul »		1
« plusieurs »		1
« n'importe quel / quelle »		5
« un tel / une telle / de tels / de telles»		5
« un/une quelconque »		5
« un/une certain/certaine »		5
« certains / certaines »		1

5.6.2. Texte automatiquement lemmatisé original

Tel que mentionné au Chapitre 4, la méthode « *PhraseStandard* » est utilisée pour générer une phrase à la fois. La méthode « *GenererTexte* » combine ensuite toutes les phrases pour former le texte complet final, qui est enregistré sur disque dans un format texte. Mais avant de générer le texte final, l'algorithme offre l'option à l'utilisateur de visualiser le détail concernant chacune des phrases à la console de l'IDE Netbeans. Cet affichage a permis un débogage plus rapide lors du développement des algorithmes. L'information apparaissant à la console est illustrée à la Figure 5.31, pour une phrase type, générée à partir des informations extraites du roman « Le Rouge et le Noir », avec les paramètres par défaut illustrés aux Tableaux 5.35 à 5.37.

```
L'image d'un quelconque progrès émerveillé se faisait des peurs patriciennes.

LE(51) IMAGE(31) DE(7) UN(51) QUELCONQUE(55) PROGRÈS(31) ÉMERVEILLÉ(2) SE(61)
FAIRE(1) UN(51) PEUR(31) PATRICIEN(2)

51*1*0  31*1*0  7  51*0*0  55*2*0  31*0*0  2*0*0  61*2*0*3*1  1*2*3  51*1*1
31*1*1  2*1*1

Article féminin singulier  ** Nom commun féminin singulier  ** Préposition  **
Article masculin singulier  ** Indéfini singulier  ** Nom commun masculin
singulier  ** Adjectif masculin singulier  ** Pronom personnel singulier 3e
pers du singulier accusatif  ** Verbe Imparfait 3e pers du singulier  **
Article féminin pluriel  ** Nom commun féminin pluriel  ** Adjectif féminin
pluriel  **
```

Figure 5.31 : Exemple de sortie à la console pour chaque phrase générée aléatoirement. La phrase elle-même apparaît sur la première ligne. Sur la deuxième ligne, on retrouve les lemmes correspondants en majuscules (pour bien les distinguer des mots de la phrase), avec le code de la classe grammaticale entre parenthèses. La troisième ligne fournit l'information morpho-syntaxique sous forme de code et la dernière ligne fournit cette même information sous forme de texte, pour chaque mot

```
L'image d'un quelconque progrès émerveillé se faisait des peurs patriciennes.
Meurs bassement plus laid! Nous irons changer la couleur d'un quelconque
conspirateur jaune, qu'un tel œil prétendit plus facilement. Nous méprisâmes
curieusement un certain semblant et un certain soulier plus mouillé, qui
faillaient à un air et à des jours très empruntés. Une quelconque docteure
pauvre, qui gazait une adversité absolument ferme, se trouvait quelques-uns.
Faites n'importe quelle cour du monsieur rapporté! Un malaise des attentions,
que l'habileté vaine et les sollicitations n'attireraient pas horriblement,
n'espérait pas redoubler de cette broderie spirituelle, que l'enchantement
produit dissimulait insensiblement. Vous te rompîtes plus machinalement. Il
fallait vivement que celle-là n'eût pas sottement répondu à une collaboratrice
presque savante. Votre grâce plus éblouie va se demander l'an apostolique, qui
croissait. Tu penseras mortellement à des déguisements vraiment abominables des
hardiesses. Une broderie spirituelle, qui n'oublia pas le partner âgé, n'eut
pas pu presque profondément précipiter votre industriel. Tu te jetas plus
hardiment ces ans. L'examinateur, que ces petites égratignures durent
paraphraser mortellement, ne reprocha pas une paysanne massacrée et une ville
isolée, qui ne faillirent pas à cette grotte. Ces institutions salutaires
n'auront pas usé fort lentement d'un document, qu'un désappointement si
singulier monnaiera. Tu passais.
```

Figure 5.32 : Exemple d'un court texte aléatoire automatiquement lemmatisé fourni en sortie

l'	le	Article féminin singulier
image	image	Nom commun féminin singulier
d'	de	Préposition
un	un	Article masculin singulier
quelconque	quelconque	Indéfini singulier
progrès	progrès	Nom commun masculin singulier
émerveillé	émerveillé	Adjectif masculin singulier
se	se	Pronom personnel 3e pers du singulier accusatif
faisait	faire	Verbe Imparfait 3e pers du singulier
des	un	Article féminin pluriel
peurs	peur	Nom commun féminin pluriel
patriciennes	patricien	Adjectif féminin pluriel

Figure 5.33 : Exemple du tableau fourni en sortie pour la première phrase de la Figure 5.32

On note qu'en plus de la phrase elle-même, l'algorithme affiche les lemmes et classes grammaticales correspondant à chaque mot, ainsi que l'information morpho-syntaxique. Mais lorsqu'on combine toutes les phrases pour former le texte, ces informations ne sont plus groupées comme à la Figure 5.31. On crée plutôt un fichier texte ne comprenant que les phrases elles-mêmes, l'une à la suite de l'autre (Figure 5.32). On constate, à la lecture des quelques phrases de la Figure 5.32, que tel que prévu, la valeur sémantique des phrases n'est pas au rendez-vous. Mais on se rappelle que là n'était pas le but premier du projet. On n'y retrouve que certaines cooccurrences. L'objectif principal est que ces phrases soient bien lemmatisées. La Figure 5.33 fournit justement l'information en lien avec la lemmatisation correspondant à la première phrase de ce court texte. On peut constater qu'effectivement, la phrase a été correctement lemmatisée. Ce tableau est enregistré sur disque dans un format facilement consultable par un chiffrier électronique comme Microsoft Excel. Un extrait du texte aléatoire final est fourni à l'Annexe I.

5.6.3. Statistiques globales pour le texte aléatoire automatiquement lemmatisé

Les statistiques globales pour le texte aléatoire automatiquement lemmatisé sont fournies elles aussi dans un fichier distinct au format texte, donc facilement lisible par n'importe quel outil de traitement de texte. Les informations fournies sont sensiblement les mêmes que pour le corpus de référence : la longueur des phrases, les lemmes les plus fréquents, les proportions de classes grammaticales, temps et personnes de verbes, et finalement les occurrences d'homographes. Ces informations sont illustrées plus bas pour un texte de 5000 phrases généré avec les paramètres par défaut listés à la Section 5.6.1, inspiré du roman « Le Rouge et le Noir ».

5.6.3.1. Longueur des phrases

Ce texte aléatoire contient 5000 phrases comprenant 77 814 mots. La Figure 5.34 fournit la distribution de la longueur des phrases, sous forme d'histogramme. On y compare les distributions de longueurs de phrases pour le corpus de référence (en gris) et le texte aléatoire (en bleu). De cet histogramme, on peut conclure que le texte aléatoire est passablement représentatif du corpus de référence en ce qui a trait aux longueurs de phrases, considérant que les formes des deux distributions sont similaires. Le roman « Le Rouge et le Noir » comporte en revanche davantage de phrases très courtes, et davantage de phrases très longues. Les phrases très courtes sont possiblement des extraits de dialogues, totalement absents du texte aléatoire. Les phrases très longues quant à elles sont probablement composées de plusieurs propositions reliées par des conjonctions, ce qu'on n'a pas tenté de simuler dans le texte aléatoire. La taille moyenne des phrases est de 17 mots pour le roman « Le Rouge et le Noir », alors que cette moyenne est de 16 pour le texte aléatoire. Là encore, on constate donc des statistiques très semblables.

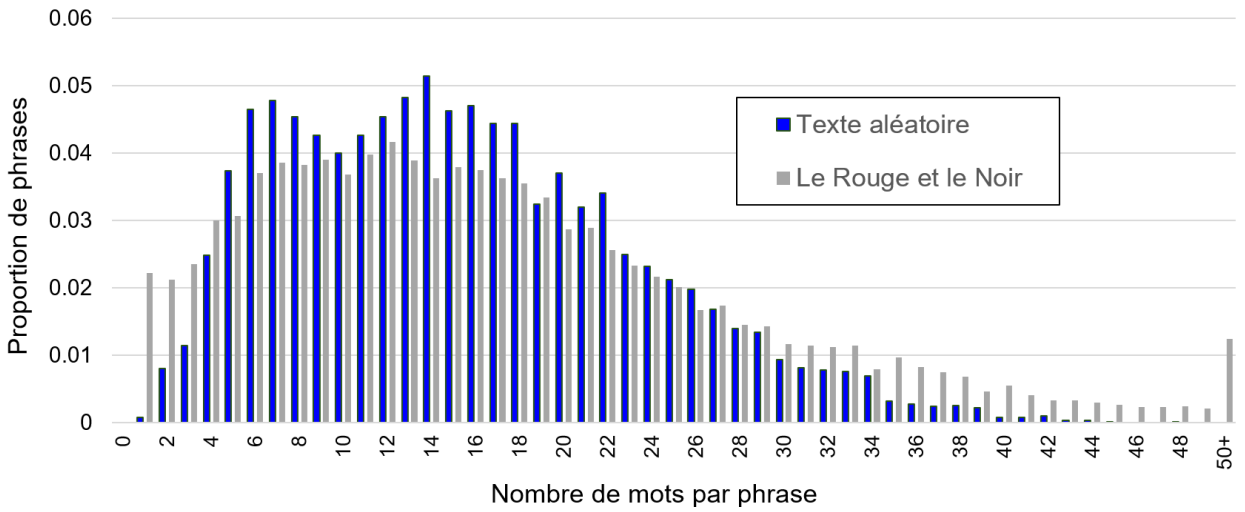


Figure 5.34 : Histogramme de la longueur des phrases du roman « Le Rouge et le Noir » (en gris) et du texte aléatoire (en bleu). L'axe horizontal représente le nombre de mots par phrase et l'axe vertical la proportion dans le texte

La longueur des phrases dans le texte aléatoire dépend de plusieurs facteurs, dont le choix des divers paramètres indiqués aux Tableaux 5.35 à 5.37. En effet, si on augmente la proportion de noms accompagnés d'adjectifs et d'adverbes, il va de soi que la taille des phrases va augmenter. Mais les deux facteurs ayant la plus grande influence sur la longueur des phrases sont la probabilité d'inclure des compléments du nom de type « du » et la probabilité d'inclure des compléments du nom de type « que ». Ces compléments du nom avaient été décrits à la Section 4.10.4.5. L'inclusion de tels compléments du nom implique l'ajout de potentiellement plusieurs mots à une phrase. Les Figures 5.35 et 5.36 illustrent justement l'effet de la présence de tels compléments du nom sur la longueur moyenne des phrases du texte généré aléatoirement (pour 5000 phrases). On y constate que la longueur moyenne des phrases passe d'environ 13 mots à environ 20 mots pour les cas extrêmes des probabilités (absence totale ou imposition systématique) pour les compléments du nom de type « du ». La différence est encore plus marquée (d'environ 11 mots à environ 20 mots) en modifiant la probabilité de présence de compléments du nom de type « que » entre les deux extrêmes.

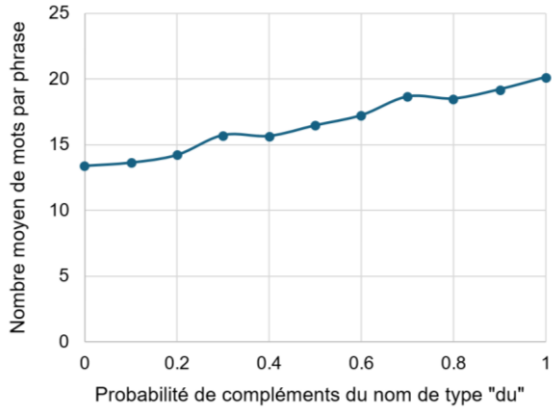


Figure 5.35 : Nombre moyen de mots par phrase en modifiant la probabilité de présence de compléments du nom de type « du »

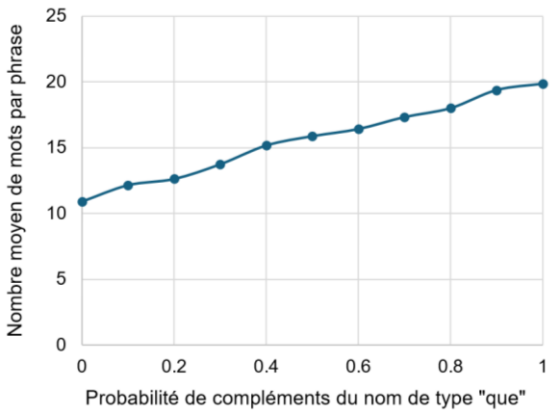


Figure 5.36 : Nombre moyen de mots par phrase en modifiant la probabilité de présence de compléments du nom de type « que »

Tableau 5.38 : Occurrences de lemmes les plus fréquents du texte aléatoire, par classe grammaticale, avec nombre d'occurrences (texte aléatoire inspiré par le roman « Le Rouge et le Noir »). Les seules classes listées sont celles inspirées par le corpus de référence (verbes, adjectifs, noms communs et adverbes)

Verbes		Adjectifs		Noms communs		Adverbes	
Lemme	#	Lemme	#	Lemme	#	Lemme	#
avoir	1312	jeune	133	œil	855	ne	3035
faillir	937	savant	126	madame	246	pas	2651
être	611	grand	100	fils	234	plus	1955
vouloir	538	brillant	89	monsieur	223	si	739
devoir	498	beau	69	homme	183	ainsi	552
pouvoir	480	fixé	68	collaborateur	175	fort	511
aller	420	étincelant	68	proie	148	presque	271
falloir	413	petit	64	femme	148	trop	255
croître	305	baissé	62	ami	146	tant	234
dire	293	moral	56	jour	141	rapidement	204

5.6.3.2. Lemmes les plus fréquents

Pour ce qui est des lemmes les plus courants dans le texte aléatoire, nous nous limitons uniquement aux verbes, adjectifs, noms communs et adverbes, puisque ce sont les seuls types de mots dont les fréquences sont guidées par celles du corpus de référence. Le Tableau 5.38 fournit donc les fréquences d'occurrence des principaux lemmes du texte aléatoire inspiré du roman « Le Rouge et le Noir », pour chacune de ces quatre classes grammaticales.

Si on compare ce tableau avec le Tableau 5.2 qui fournissait cette même information pour le corpus de référence, on note que 6 des 10 verbes sont communs aux deux listes, soit les verbes « avoir », « être », « vouloir », « pouvoir », « aller », et « dire », bien que leurs rangs respectifs diffèrent quelque peu. Il est à noter que tous les verbes du Tableau 5.38 sont du troisième groupe. Pour ce qui est des adjectifs, on en note quatre en commun avec le roman original (« jeune », « grand », « beau » et « petit »). On note que deux des adjectifs du tableau sont en fait des participes passés (« fixé » et « baissé »), alors que deux autres peuvent aussi être des participes présents (« brillant » et « étincelant »). Au niveau des noms communs, on note six mots en commun au tableau en comparaison avec le roman « Le Rouge et le Noir », soit « œil », « madame », « monsieur », « homme », « femme » et « jour ». Pour ce qui est des adverbes, on en note cinq en commun. Tel que mentionné précédemment, les adverbes les plus fréquents, au même titre que les verbes, n'offrent pas d'indices particuliers sur l'origine du corpus de référence, puisqu'ils sont assez génériques. La plupart des verbes et adverbes listés ici sont vraisemblablement communs à tout texte de la langue française. En général, les similitudes observées entre le Tableau 5.38 pour le texte aléatoire et le Tableau 5.22 pour le roman « Le Rouge et le Noir » témoignent du fait que le texte aléatoire s'est effectivement inspiré des fréquences du corpus de référence.

Si on laisse de côté les lemmes en particulier et qu'on s'attarde plutôt à la proportion globale des différentes classes grammaticales, on observe encore des similitudes entre le corpus de référence et le texte aléatoire (Tableau 5.39). En effet, les proportions de verbes (17% vs. 19%), de noms (22% vs. 17%) de déterminants (16% vs. 16%) et de pronoms (13% vs. 11%) sont très semblables. On note cependant des proportions environ doubles pour les adjectifs et les adverbes dans le texte aléatoire. On pourrait toutefois facilement diminuer ces fréquences dans le texte

aléatoire en diminuant les paramètres de probabilités pour adjectifs et adverbes listés au Tableau 5.35. On observe toutefois beaucoup moins de prépositions (le tiers) et moins de conjonctions dans le texte aléatoire. Ces valeurs réduites pour le texte aléatoire témoignent du fait que les structures de phrases générées dans ce texte sont bien plus simples que celles du roman « Le Rouge et le Noir ». Finalement, aucun effort n'a été fait pour introduire des interjections au texte aléatoire. Il est donc tout à fait normal que cette proportion y soit de zéro.

5.6.3.3. Personnes et temps de verbes

Le Tableau 5.40 illustre les proportions des personnes de verbe. On y retrouve d'une part les proportions recherchées, sur la base de ce qu'on retrouvait dans le corpus de référence, mais modifiées pour assurer une présence minimale de chaque temps. Ces informations avaient été fournies au Tableau 5.31. On compare ensuite ces proportions à celles retrouvées dans le texte aléatoire. On note que pour la troisième personne (surtout au pluriel), les proportions sont plus grandes pour le texte aléatoire. Ceci s'explique par l'imposition de la troisième personne dans les compléments du nom de type « que » (Section 4.10.4.5), car on a arbitrairement décidé de n'inclure à ces compléments que des noms comme sujets, et aucun pronom. En imposant des probabilités nulles de compléments du nom de type « que », on retrouverait les proportions de personnes recherchées.

Le Tableau 5.41 illustre les proportions des temps de verbe. On y retrouve d'une part les proportions recherchées, sur la base de ce qu'on retrouvait dans le corpus de référence, mais modifiées pour assurer une présence minimale de chaque temps. On y retrouve aussi les proportions observées dans le texte aléatoire. D'importantes différences sont notées entre les pourcentages des deux colonnes du Tableau 5.41. Mais il faut préciser que les proportions recherchées (première colonne) n'ont servi qu'à déterminer le temps du *verbe principal du groupe du verbe*. Plusieurs autres verbes se greffent ensuite à la phrase, sans que le temps de ces verbes soit dicté par ces proportions recherchées. Par exemple, à chaque fois qu'un verbe modal est utilisé (« devoir », « pouvoir », etc.), on suit avec un verbe à l'infinitif (« je dois *manger* »). De la même façon, à la suite des prépositions « à », « de », « pour » et « sans » au groupe du complément, on introduit aussi un infinitif (« je pense à *manger* »). Ces cas expliquent la prominence de l'infinitif (16%) dans le texte aléatoire, alors qu'aucun seuil n'avait été dicté pour ce temps. Ainsi, toutes les autres proportions se retrouvent affectées.

Tableau 5.39 : Proportions des mots appartenant à chaque classe grammaticale, pour le roman « Le Rouge et le Noir » (après désambiguïsation) et le texte aléatoire de 5000 phrases

Classe grammaticale	Le Rouge et le Noir	Texte aléatoire
Verbes	17%	19%
Adjectifs	6.7%	10%
Noms	22%	17%
Adverbes	6.9%	17%
Déterminants	16%	16%
Pronoms	13%	11%
Prépositions	13%	5.4%
Conjonctions	4.6%	3.3%
Interjections	0.12%	0.0%

Tableau 5.40: Proportions des personnes de verbe inspirées du corpus de référence mais modifiées pour assurer une présence minimale de chacune, et proportions pour le texte aléatoire

Personnes de verbe	Proportions des personnes de verbe	
	Recherchées	Texte aléatoire
1ère du singulier	10%	7.8%
2e du singulier	10%	13%
3e du singulier	50%	51%
1ère du pluriel	10%	4.9%
2e du pluriel	10%	5.4%
3e du pluriel	10%	18%

Tableau 5.41: Proportions des temps de verbe inspirées du corpus de référence mais modifiées pour assurer une présence minimale de chacun, et proportions pour le texte aléatoire

	Proportions de temps de verbe	
Temps de verbe	Recherchées	Texte aléatoire
Infinitif	0%	16%
Présent	19%	7.0%
Imparfait	32%	21%
Passé simple	22%	14%
Futur simple	5.0%	1.8%
Subjonctif présent	2.0%	0.65%
Subjonctif imparfait	2.0%	0.72%
Impératif	2.0%	0.68%
Conditionnel	2.0%	1.4%
Participe présent	0.0%	0.0%
Participe passé	0.0%	27%
Passé composé	2.0%	2.2%
Plus-que-parfait	2.0%	4.3%
Passé antérieur	2.0%	1.1%
Futur antérieur	2.0%	1.1%
Subjonctif passé	2.0%	0.67%
Subjonctif plus-que-parfait	2.0%	0.77%
Conditionnel passé	2.0%	1.1%

5.6.3.4. Présence d'homographes

Il faut maintenant se rappeler qu'un des objectifs principaux de ce projet est de générer des textes aléatoires qui vont confronter des lemmatiseurs existants à la difficulté de lemmatiser des homographes. On avait d'abord noté que 32% des mots du corpus de référence étaient des homographes. Dans le texte aléatoire, on parle de 33% (25 797 homographes sur 77 814 mots). Ce pourcentage est donc très semblable à la proportion retrouvée dans le corpus de référence. On peut donc en conclure que le texte aléatoire comprend un nombre bien suffisant d'homographes auxquels confronter les outils de lemmatisation existants.

Tout comme on l'a fait pour le corpus de référence (Tableaux 5.1 et 5.2), deux tableaux d'homographes sont fournis pour le texte aléatoire. Le Tableau 5.42 fournit les quantités d'homographes distincts dans le texte aléatoire, tandis que le Tableau 5.43 fournit la fréquence de ces mêmes homographes.

Tableau 5.42 : Quantité d'homographes *distincts* dans le texte aléatoire de 5000 phrases, en fonction des classes grammaticales impliquées

	Verbes	Adjectifs	Noms	Adverbes	Déterminants	Pronoms	Prépositions	Conjonctions	Interjections
Verbes	59	1346	694	2	0	3	2	0	0
Adjectifs	1346	5	509	7	9	4	0	0	0
Noms	694	509	6	7	2	1	4	0	2
Adverbes	2	7	7	0	0	0	0	1	0
Déterminants	0	9	2	0	0	12	2	0	0
Pronoms	3	4	1	0	12	5	0	3	0
Prépositions	2	0	4	0	2	0	0	0	0
Conjonctions	0	0	0	1	0	3	0	0	0
Interjections	0	0	2	0	0	0	0	0	0

Tableau 5.43 : Quantité d'homographes dans le texte aléatoire de 5000 phrases, en fonction des classes grammaticales impliquées

	Verbes	Adjectifs	Noms	Adverbes	Déterminants	Pronoms	Prépositions	Conjonctions	Interjections
Verbes	665	5811	4290	1888	0	19	7	0	0
Adjectifs	5811	78	3661	607	537	68	0	0	0
Noms	4290	3661	116	3095	234	13	14	0	16
Adverbes	1888	607	3095	0	0	0	0	875	0
Déterminants	0	537	234	0	0	6339	2638	0	0
Pronoms	19	68	13	0	6339	3553	0	1714	0
Prépositions	7	0	14	0	2638	0	0	0	0
Conjonctions	0	0	0	875	0	1714	0	0	0
Interjections	0	0	16	0	0	0	0	0	0

En s'attardant en premier lieu à la diagonale de ces deux tableaux, on constate que 3.5% des homographes distincts du texte aléatoire sont issus de la même classe grammaticale. Ce chiffre était de 3.0% pour le corpus de référence, deux valeurs assez semblables. Si on s'attarde aux fréquences des homographes, on obtient aussi un résultat très semblable pour les deux textes. En effet, on observe un pourcentage de 17% d'homographes de même classe grammaticale pour le texte aléatoire, en comparaison avec 18% pour le corpus de référence. On peut donc en conclure que le texte aléatoire a généré suffisamment d'homographes de même classe grammaticale. Il faut se rappeler que de tels homographes sont potentiellement plus difficiles à désambiguïser que ceux impliquant des classes grammaticales différentes. Il était donc essentiel d'en inclure un nombre suffisant dans le texte aléatoire.

En comparant les résultats pour les deux types de texte, on constate que dans les deux cas, les homographes *distincts* les plus fréquents sont du type « verbes-adjectifs » (63% pour le texte aléatoire, 52% pour le corpus de référence). Ce pourcentage élevé est dû au fait que tous les participes passés dans ce projet sont aussi par défaut considérés comme des adjectifs (participes passés employés seuls). La fréquence plus élevée de ce type d'homographes dans le texte aléatoire est due au paramètre élevé pour la probabilité de présence d'adjectifs.

Si on tient compte des fréquences d'apparition, les deux types de texte donnent aussi des résultats semblables pour le type d'homographe le plus courant, soit le type « déterminant-pronom » (25% pour le texte aléatoire, 32% pour le corpus de référence). Les homographes « le », « la » et « les » sont les plus fréquents de ce type.

Il importe aussi de mentionner qu'à l'exception de ceux impliquant les conjonctions et interjections (très rares de toute façon), le texte aléatoire a pu engendrer des homographes pour presque toutes les paires de classes grammaticales pour lesquelles on en avait retrouvés dans le corpus de référence. Les exceptions sont les paires « adverbe-déterminant », « adverbe-pronom » et « adjectif-préposition ». Mais ces cas représentent moins de 1% des cas d'homographes distincts.

5.6.4. Salade de mots

Pour générer la salade de mots, on modifie complètement l'ordre des *mots* du texte aléatoire, indépendamment des phrases auxquelles ils appartenaient. Aucun effort n'est fait pour reséquencer les mots d'une certaine façon. En effet, bien au contraire, on fait appel à une procédure basée sur un pur hasard. Tel que mentionné aux Sections 3.5.6 et 4.10.8, on s'attend donc à se retrouver avec des « phrases » non seulement vides de sens, mais aussi sans queue ni tête, car aucune règle syntaxique n'y est imposée.

Comme la salade de mots comprend exactement les mêmes mots que le texte aléatoire original, les statistiques globales demeurent les mêmes pour la salade de mots. Il est donc inutile d'une fois de plus discuter de la fréquence des lemmes, des proportions de classes grammaticales et de la présence des homographes. Ces résultats, discutés à la section précédente pour le texte aléatoire original, seraient identiques pour la salade de mots. Nous allons donc plutôt nous limiter ici à une brève analyse *qualitative* de cette salade.

La Figure 5.37 fournit en exemple les quelques premières phrases de ce texte reséquencé, toujours pour le même exemple de texte aléatoire comportant 5000 phrases. Rappelons-nous, tel que mentionné à la Section 4.10.8, que des « phrases » ont été artificiellement créées au sein de cette salade de mots, en faisant coïncider le début et la fin des phrases (en termes de nombre de mots) avec celles du texte aléatoire original. Un plus long extrait de cette salade de mots est fourni à l'Annexe I.

```
Cette les premier le prodige chêne une accabla formation là madame
tel étranger assista n'une dus qui les prudemment désirs une. Qui ne
d'aux leurs quelconque ai d'avancements ne à liberté ainsi ne. Ces
doucement faisait pas profiter baptiser allés ambition cette. De des
je vous première des ne si avancement osent qu'accueils des. Progrès
telle rassurées une dit répondait vivement plus publics à vivement
sages n'infamie ne que main qui sûretés mépris les ainsi nues jour
tel ses qui rapprochée aura qui cette louis là plus très ce parlait
nouvelle de. De premières marmots importe humaines vrais n'un à
propreté imprévues pas rien voix cette parfaitement altier des
premières fasse. Qui sérieuse. Que de à vrais conquêtes un récit une
ne. Plus des un ne ces s'a trouvez suppliantes et plus normandes et
et accoutrement chirurgiennes refouler irrésistible rendues pas exposé
pas paresseuses.
```

Figure 5.37 : Exemple de « salade de mots » fourni en sortie. Seules les premières « phrases » des 5000 phrases générées sont affichées, par souci de concision. Le début et la fin des « phrases » coïncide avec ce qu'on retrouve dans le texte aléatoire original

Tel qu'anticipé, on se retrouve effectivement avec un texte sans queue ni tête. Mais malgré cette absence de sens ou de structure, cette salade de mots sera utile lorsque viendra le temps d'évaluer les outils de lemmatisation existants, en les confrontant à ce genre de texte, au Chapitre 6. En effet, la performance de lemmatisation de la salade de mots pourra être comparée avec celle pour le texte aléatoire original. Une absence de différence indiquerait que l'outil de lemmatisation ne tient pas compte de l'analyse syntaxique.

5.6.5. Étude paramétrique

Tel que mentionné à la Section 5.6.1, plusieurs paramètres ont dû être déterminés relativement arbitrairement pour la génération de phrases au hasard. Les Tableaux 5.35 à 5.37 ont inclus les valeurs de ces paramètres utilisés pour générer le texte utilisé comme exemple de sortie de l'algorithme actuel. L'influence de certains de ces paramètres a aussi été discutée lors de l'analyse des statistiques globales du texte aléatoire (Section 5.6.3), et l'effet de deux de ces paramètres en lien avec les compléments du nom a été étudié dans le contexte de la longueur moyenne des phrases (Section 5.6.3.1). Il demeure toutefois pertinent d'aller un peu plus loin, et d'observer l'influence de la modification de certains de ces paramètres sur les textes générés et leurs caractéristiques associées.

Mais en plus de ces paramètres, d'autres facteurs exercent une influence certaine sur les textes fournis en sortie. Par exemple, l'utilisation de cooccurrences, la sélection du corpus de référence et le processus de désambiguïsation des homographes. Une étude paramétrique limitée a donc été réalisée pour étudier l'effet de certains paramètres des Tableaux 5.35 à 5.37 ainsi que celui de ces facteurs additionnels.

5.6.5.1. Effet de la modification de certains paramètres

Quelques modifications sont d'abord apportées à la valeur de quelques paramètres du Tableau 5.35 pour analyser l'impact de ces changements sur les phrases générées, dans le but d'une part s'assurer de cette fonctionnalité, puis de considérer qualitativement l'effet observé. Plus spécifiquement, les changements suivants vont être indépendamment apportés :

- Proportion de verbes modaux : le paramètre passe de 0.35 (défaut) à 1.0 (utilisation exclusive de verbes modaux)

- Proportion de formes verbales négatives : le paramètre passe de 0.3 (défaut) à 1.0 (utilisation exclusive de formes négatives)
- Proportion de formes avec noms dans les groupes du sujet et du complément : le paramètre passe de 0.8 (défaut) à 0.0 (utilisation exclusive de pronoms)
- On imposera aussi le participe passé comme temps de verbe principal (ce paramètre ne figure pas au Tableau 5.35).

Les Figures 5.38 à 5.41 fournissent quelques phrases générées avec les modifications apportées à ces quatre paramètres. L'analyse de ces phrases confirme en effet que les contraintes imposées ont bel et bien été respectées par l'algorithme, avec l'utilisation exclusive de verbes modaux à la Figure 5.38 et celle de formes verbales négatives à la Figure 5.39, l'utilisation exclusive de pronoms à la Figure 5.40 et finalement l'imposition du passé composé comme verbe principal. Dans le cas de la Figure 5.41, on peut en profiter pour constater que les règles d'accords des participes passés ont été respectées dans toutes les phrases.

Dans tous ces tests paramétriques, des valeurs extrêmes ont été imposées pour facilement s'assurer que les contraintes aient été respectées. Mais surtout, ces résultats démontrent que les paramètres peuvent être modifiés à la guise des utilisateurs pour atteindre les buts qu'ils poursuivent.

```
L'œil fixé, qui alla timidement faillir les nôtres, devait dire gaiement
des académiciennes, qu'une certaine marquise plus fanatique peut avouer.
Je pouvais quitter un canapé bleu. Tu n'allais pas assurer le rang de
leur lustre plus troisième, qui allait faillir plus profondément une
amie piémontaise. Un certain ami, auquel les marquises domptées
voulurent faillir, pourra se dire ainsi celui-ci. L'animal fort déposé,
qui ne dut pas croire autrement, ne pensera pas parler du nôtre. Tu
allais simplement torturer une bonne si provinciale de ce désintérêt
plus mécontent. Le ménage de telles formalités folles, auquel le
déboursé si annuel a dû faillir, semble penser légalement à la
hardiesse. Une insensibilité brutale alla inspirer ceux-là. Un œil
hagard de l'œil étincelant, que le narrateur et l'azur durent ouvrer,
voulait avouer si ainsi une quelconque larme baignée. Nous ne voulions
pas nous renfermer votre veste.
```

Figure 5.38 : Phrases aléatoires générées en imposant une proportion de verbes modaux de 1.0 (100%)

```
Mon patelin contrit, qui ne blasphémait pas la galère, ne prononçait
pas les pruderies modernes. Les animaux déposés et la mansarde des
hameaux habités n'étaient pas allés descendre. Ma récidive des soies,
qui ne croissait pas, n'était pas allée retourner. Le parti, qui ne
croissait pas rapidement, n'avait pas parfaitement pensé prendre la
concession. Il ne pourra pas s'agir de quelques vergognes, qui
n'influaient pas si proprement sur tes sous-directrices nécessaires et
sur ton œil fixé. Il ne fallait pas ainsi que cette marquise, à laquelle
un monsieur continuel ne pense pas faillir, n'eût pas simplement dit la
sous-préfète galante. Celui-ci ne faisait pas trop régulièrement de
peurs et d'efforts du cadeau presque magnifique, qui ne faillirent pas
sur ce progrès rapide. Je ne pleurai pas les vieillards plus corrompus
de leur œil. Il fallait sincèrement que je n'aime pas fort profondément
importer l'œil.
```

Figure 5.39 : Phrases aléatoires générées en imposant une proportion de formes verbales négatives de 1.0 (100%)

```
Il n'aimait pas simplement tirer ceux-là. Tu nous levais certainement.
Celui-ci allait si parfaitement admirer le nôtre. Vous vouliez les
tiens. La nôtre choisit plus parfaitement quelqu'un. Nous nous dîmes
horriblement les vôtres. J'appris les nôtres. Celui-là ne s'avance pas
plus matériellement la tienne. Commencez le leur! Celle-là s'aperçut
des nôtres. La nôtre a voulu plus savamment le tien. Tu voulus remercier
rapidement la nôtre. La nôtre put le vôtre. Tu n'entres pas dans les
siennes. Je me rendais horriblement à celle-là. Les siens ne faisaient
pas ceux-là. Elle est allée perdre quiconque. Le nôtre va se réjouir de
celles-ci. Je me serai trouvé fort mortellement le vôtre. Vous vous
pénétriez la nôtre. Vous la destituiez certainement. Celle-ci voudra
mépriser les nôtres. Je ne pouvais pas traverser tant timidement les
tiennes.
```

Figure 5.40 : Phrases aléatoires générées en imposant l'utilisation exclusives de pronoms (aucun nom) dans les groupes du sujet et du complément

```
Des apparences des yeux baissés ont si simplement pu se figurer diverses
joues pâles, que l'œil pourpre ne put pas précisément dessiller. Leurs
normandes montantes de vos chevaux n'ont pas pu consentir à ce poste
des facteurs et d'une belle possesseure. Tu es allé ôter quelqu'un. Il
s'est agie des doses et du ressort mis. Un certain logement s'est refusé
le nôtre. Vous avez semblé reprendre diverses éditrices et diverses
très grandes haleines d'une telle dame, qui ne renouvelèrent pas ceux-
là. Tu es entrée dans la péruvienne. Tu as répété l'assurance favorable.
Tu as voulu distinctement daigner cette diplomate d'une belle
demoiselle, que ces yeux plus étincelants achetèrent adroitement. J'ai
pu m'approcher ordinairement des plaidoiries si décisives et d'une
volée. Vous vous êtes crus des machiavels plus effrénés. Un quelconque
bijou appuyé, qui soufflait tes chiffons plus provinciaux, a regardé
quelques messieurs, qui ne croissent pas.
```

Figure 5.41 : Phrases aléatoires générées en imposant l'utilisation exclusive du participe passé comme *verbe principal du groupe du verbe*

5.6.5.2. Effet de l'incorporation de cooccurrences

Lors de la lemmatisation du corpus de référence, un algorithme s'est chargé de comptabiliser les cooccurrences pour chacun des lemmes de verbes, adjectifs, noms communs et adverbes. Ces cooccurrences consistent en une liste de lemmes les plus souvent retrouvés dans l'environnement immédiat du lemme en question. Le but de cette opération, tel que mentionné à la Section 3.3, est d'ensuite tenter d'injecter un minimum de sens aux phrases aléatoires en tenant d'y reproduire ces mêmes cooccurrences. Toutefois, comme on l'a vu à la lecture du texte généré aléatoirement à la Figure 5.32, le résultat ne s'est pas avéré très convaincant. En effet, la sémantique n'était pas au rendez-vous. Toujours est-il qu'il est intéressant de voir de quoi peut avoir l'air un texte inspiré du même corpus de référence (le roman « Le Rouge et le Noir ») mais n'incorporant pas de cooccurrences. La Figure 5.42 fournit quelques phrases ainsi générées sans appel aux cooccurrences. Là encore, on retrouve des phrases totalement dépourvues de sens. Mais en comparant ce texte avec celui de la Figure 5.32, il semble que le texte dépourvu de cooccurrences a encore moins de sens. C'est donc peut-être à la lecture du texte de la Figure 5.42 qu'on peut finalement apprécier le fait que les cooccurrences ont après tout quelque peu aidé à injecter un minimum de sens aux phrases.

```
Les causes et la madame, qui ne pensent pas vivement l'empire
malheureux, achetaient des temps choquants et des propriétaires
excellentes, qui disparaissent gaiement. Un tabac fort fin de ces
témoins et de ces amies doit naturellement étendre des rapidités utiles.
Notre ami méprisable et notre sang, qui n'auraient pas vu les désespoirs
trop lents et la marquise, se disaient vivement ce voisin plus étonné
de vos milieux et de votre lendemain étourdi, que tes sujets presque
innombrables prouvaient ainsi. Vous ne vous crachâtes pas les nôtres.
Une telle société bleue ne devait pas avoir plus vivement cet an. Un
fait fermé pouvait sincèrement persuader la forme, que des actrices
voyaient. Un quelconque courage nécessaire, qui voyagea, rêve de faire
des seigneurs fort grillés et un homme plus séparé, qui voyaient
fixement une note consolée. Un zèle plus cruel n'était pas naturellement
allé se faire cet instant rassuré, qui avait détesté profondément la
cour.
```

Figure 5.42 : Phrases aléatoires générées en ne considérant pas les cooccurrences

5.6.5.3. Effet de la sélection du corpus de référence

Le corpus de référence est à la base des banques lexicales de verbes, noms communs, adjectifs et adverbes parmi lesquelles l'algorithme de génération de phrases sélectionne ces termes, en fonction de leurs fréquences d'occurrences. C'est aussi à partir du corpus de référence qu'on extrait les fréquences de temps de verbe et les personnes de verbes présents dans le corpus, sous forme de plages de valeurs pour chacun de ces temps. Finalement, le corpus de référence sert aussi à bâtir les tableaux de cooccurrences, qui inclut la fréquence d'occurrences de paires de mots se retrouvant dans l'environnement immédiat. Il va donc de soi que le corpus de référence influence de multiples façons le contenu des textes générés aléatoirement à l'Étape 2 du projet.

Afin de constater le rôle prépondérant du corpus de référence, un texte aléatoire a été créé en s'inspirant cette fois du roman de science-fiction décrit plus tôt, afin de le comparer à celui inspiré du roman « Le Rouge et le Noir » discuté précédemment à la Section 5.6.3. Une analyse complète de ce texte ne sera pas effectuée. On se contentera d'une brève analyse qualitative du texte généré, affiché à la Figure 5.43.

À la lecture de ce court extrait, on note une certaine différence avec le vocabulaire utilisé précédemment pour le texte issu du roman « Le Rouge et le Noir », ce qui confirme que les mots choisis sont guidés par les fréquences d'apparition des mots dans le corpus de référence. Mais surtout, on note dans ce nouveau texte une bien plus grande prépondérance du présent de l'indicatif. On retrouvait au contraire beaucoup de phrases à l'imparfait et au passé simple dans le texte inspiré du roman « Le Rouge et le Noir », suivant le style adopté par Stendhal, l'auteur de ce roman. On constate donc que tel que prévu, les textes aléatoires s'inspirent du corpus de référence pour favoriser certains temps de verbes.

Le fait que les textes aléatoires soient ainsi influencés par le corpus de référence, autant pour le lexique que pour les caractéristiques de verbes, témoigne de la pertinence pour ce projet, de choisir un corpus de référence approprié pour les textes à lemmatiser subséquemment.

```
Il pourfend mon souvenir extraordinaire d'un marchand terrien. Tu t'essaies une
solution plus inopinée de l'heure. Tu as tapé ce coéquipier d'une quelconque
allégresse impromptue. Quiconque agglutine des voyageuses trop vêtues d'un
quelconque terminal. Tu déduis directement sa technocrate trop assignée et ses
routeurs, que la donnée et les sécurités ne voudraient pas humblement référencer.
Des gardes te salissaient. Une certaine enfance, à laquelle une partisane doit
remédier, a voulu se fier une selle si présente et une hécatombe deuxième, qui
ne croissent pas. Il ne faut pas de tentacules très rassurantes et de
mercenaires. Celui-là doit régler ce problème plus technique d'un tel leurre
trop fort. Chaque raison évalue plus péniblement la banquette des heures plus
défraichies, à laquelle cette tentacule a failli. Un tel signe très exaspéré et
de telles médecins ne savent pas faire nul applaudissement bruyant. Nous voulons
louper une telle conversation, que ces milieux parsemés et ces diatribes ne
captent pas. N'importe quelle acolyte de tes disciples, qui doit déloger ton
temple, s'est rejointe une difficulté plus atteinte de ces balais et de ce mètre
rectangulaire. Il fallait fort autrement que vous eussiez dû vous réjouir de
ces tentacules, à laquelle un quelconque moule secret faut. Des ministres si
coloniaux de l'administration fort opposée s'étonnent d'une bedaine plus
élégante et des dissimulations.
```

Figure 5.43 : Exemple de texte aléatoire automatiquement lemmatisé fourni en sortie en utilisant le roman de science-fiction comme corpus de référence. Seules quelques phrases sont affichées, par souci de concision

5.6.5.4. Effet de la désambiguïsation des homographes

La désambiguïsation des homographes n'est qu'un objectif secondaire de ce projet, mais ayant tout de même une influence sur les résultats, puisqu'on évite par ce processus d'assigner des fréquences trop élevées pour certains lemmes incorrectement identifiés ou assignés à la mauvaise classe grammaticale. Il faut en effet se rappeler que dans le cas où on n'effectue pas de désambiguïsation, on ne peut déterminer avec certitude le lemme et la classe grammaticale de la grande majorité des homographes.

Afin de vérifier le bénéfice d'avoir inclus ce processus de désambiguïsation, il suffit de nous attarder à certains homographes en particulier, et à comparer leurs fréquences estimées sans et avec désambiguïsation. L'homographe « plus » s'avère pertinent pour cet exercice, surtout au sein du roman de science-fiction. L'homographe « plus » peut être soit la forme conjuguée du verbe « plaire » aux première et deuxième personnes du passé simple (« tu me plus »), ou plutôt un adverbe (« je ne mange *plus* », « il est *plus* beau »). Mais dans le roman de science-fiction, l'homographe « plus » n'est *jamais* utilisé comme une forme conjuguée du verbe « plaire » (le passé simple n'est que très peu utilisé dans ce texte).

En supposant une lemmatisation parfaite du roman de science-fiction, impliquant une désambiguïsation parfaite des homographes, on devrait donc s'attendre à ne jamais retrouver le lemme « plaire » associé à l'homographe « plus ». Au contraire, si on ne procède pas à la désambiguïsation des homographes, toutes les possibilités de lemmes pour l'homographe « plus » seront conservées et utilisées lors de la compilation de leurs fréquences respectives. Il en découle qu'on retrouvera une plus grande fréquence du verbe plaire dans le texte aléatoire, qu'on devrait retrouver à la suite d'une désambiguïsation efficace.

Le Tableau 5.44 résume l'analyse effectuée. On y constate que l'homographe « plus » apparait 219 fois dans le roman, et qu'à chaque fois, il s'agit de l'adverbe et non de la forme verbale. Lorsqu'on ne procède pas à la désambiguïsation, l'algorithme attribue l'homographe « plus » autant aux deux formes possibles (verbe et adverbe), ne pouvant les distinguer. On voit donc la même fréquence de 219 assignée aux deux possibilités.

Tableau 5.44 : Fréquences d'apparition de l'homographe « plus » dans le roman de science-fiction

Lemme (classe)	Fréquences d'apparition				
	Réelles dans le corpus	Sans désambiguïsation		Avec désambiguïsation	
		Attribuées au sein du corpus	Insérées dans le texte aléatoire	Attribuées au sein du corpus	Insérées dans le texte aléatoire
« plaire » (verbe)	0	219	2	0	0
« plus » (adverbe)	219	219	1114	219	2523

On note ensuite qu'au sein des 5000 phrases générées aléatoirement, on a retrouvé deux occurrences du verbe « plaire », dues à la fréquence non nulle attribuée au lemme « plaire » par l'algorithme de lemmatisation qui n'a pas encore procédé à la désambiguïsation des homographes. En revanche, quand on procède à la désambiguïsation, les 219 occurrences de « plus » sont toutes correctement associées à la forme adverbiale. Il en résulte que le texte aléatoire ne contient dans ce cas aucune occurrence du verbe « plaire », ce qui est en accord avec le corpus de référence qui n'en contient pas non plus. Une analyse semblable pourrait être effectuée pour d'autres homographes, pour démontrer la pertinence de la désambiguïsation.

5.6.6. Information fournie en sortie

Les résultats fournis en sortie par l'algorithme de l'Étape 2 peuvent être résumés ainsi :

- Le texte aléatoire automatiquement lemmatisé
- Pour ce texte :
    - Un tableau de trois colonnes incluant : chaque mot, son lemme correspondant, son information morpho-syntaxique
    - Un fichier de statistiques globales
- Un « texte » représentant la « salade de mots »
- Pour cette salade de mots :
    - Un tableau de trois colonnes incluant : chaque mot, son lemme correspondant, son information morpho-syntaxique
    - Un fichier de statistiques globales

Ce sont donc ces deux textes auxquels on soumettra les outils de lemmatisation existants au Chapitre 6, et c'est sur la base de l'information incluse aux tableaux d'information cités plus haut qu'on évaluera l'efficacité de ces outils. À l'Annexe I, on fournit un plus long échantillon de ces deux textes ainsi que des extraits des tableaux de trois colonnes pour le texte aléatoire automatiquement lemmatisé, ainsi que pour la salade de mots.

5.6.7. Vitesse d'exécution de la génération de textes

Tel que précisé à la Section 5.5, la rapidité d'exécution n'est pas un objectif principal de ce projet, mais elle demeure tout de même souhaitable. Le but de cette section est donc de donner une idée de la vitesse d'exécution de l'Étape 2 de ce projet, qui comprend la génération des phrases aléatoires, la génération d'une salade de mots, et finalement la sauvegarde de fichiers comprenant les textes et tableaux de résultats.

Tel que mentionné à la Section 5.5, le temps d'exécution a été mesuré grâce à la fonction « *System.currentTimeMillis()* » de Java. Pour évaluer ces temps d'exécution, l'algorithme de l'Étape 2 a été exécuté pour un total de 5000 phrases aléatoires en faisant le suivi du temps requis

pour générer chacune d'entre elles. La Figure 5.44 offre en premier lieu un histogramme de la distribution du temps de génération des phrases individuelles. On note qu'environ le tiers des phrases ont été générées en 50 millisecondes ou moins alors qu'environ 3% d'entre elles ont nécessité plus d'une seconde chacune. Ces écarts s'expliquent par les différences de complexité de chacune des phrases et le nombre de mots qu'elles contiennent. Par exemple, une phrase comprenant un verbe intransitif conjugué à l'impératif peut se retrouver à ne contenir qu'un seul mot (« Viens! »). À l'autre extrême, comme on l'a vu à la Section 5.6.2, certaines phrases peuvent faire intervenir divers compléments du nom requérant plusieurs tests pour les accords grammaticaux, et contenant quelques dizaines de mots. De telles longues phrases nécessitent plus de temps à se construire.

Le facteur principal influençant le temps d'exécution global de l'Étape 2 est le nombre de phrases à générer. Ce nombre de phrases est laissé libre à l'utilisateur selon ses besoins. Ce facteur est donc varié pour en étudier l'influence sur le temps d'exécution, ce qui permet de déterminer l'ordre algorithmique de ce processus de génération de phrases aléatoires. Tel que précisé à la Section 5.5, le temps d'exécution dépend forcément aussi de la vitesse et de l'efficacité de l'ordinateur utilisé. Mais le rôle de la machine utilisée n'étant pas pertinent ici, cette variable est ignorée en mettant l'emphase sur les temps d'exécution *relatifs*.

Chaque phrase est générée indépendamment des autres, si bien qu'on doit s'attendre à un ordre logarithmique linéaire ($\Theta(n)$). Autrement dit, si on double le nombre de phrases à générer, on doit s'attendre en moyenne à aussi doubler le temps d'exécution. Pour la génération de fichiers texte en sortie, on peut là encore s'attendre à un ordre logarithmique linéaire, puisqu'en doublant le nombre de phrases, on double la longueur des textes et des tableaux, bien que le fichier de statistiques globales conserve la même taille.

Ces temps d'exécution sont illustrés graphiquement à la Figure 5.45 où l'axe horizontal représente le nombre de phrases générées et l'axe vertical représente le temps d'exécution en secondes. On constate, tel qu'anticipé, que les temps d'exécution sont très fidèles à un ordre logarithmique linéaire.

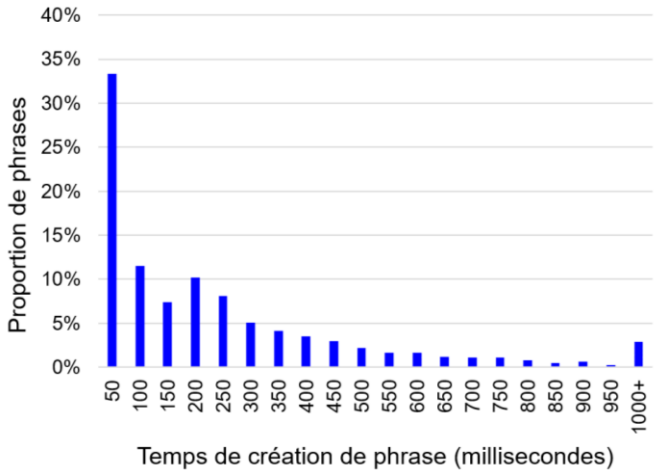


Figure 5.44 : Histogramme des temps de création de phrases aléatoires. L'appareil utilisé est un ordinateur personnel Acer Aspire X3470 muni d'un processeur AMD A6-3620 APU de 2.20 GHz

Figure 5.45 : Temps d'exécution pour la création de phrases aléatoires. L'appareil utilisé est un ordinateur personnel Acer Aspire X3470 muni d'un processeur AMD A6-3620 APU de 2.20 GHz

6. ÉVALUATION D'OUTILS DE LEMMATISATION EXISTANTS

Tout est maintenant en place pour passer à l'étape suivante, qui vise à évaluer le plus objectivement possible la performance d'outils de lemmatisation existants, sur la base, entre autres, des textes aléatoires automatiquement lemmatisés générés dans le cadre de ce projet. Cette étape est l'étape ultime du projet, celle pour laquelle tous les efforts déployés aux Étapes 1 et 2 convergent. En effet, la lemmatisation du corpus de référence à l'Étape 1 n'a servi qu'à fournir les données nécessaires à l'exécution de l'Étape 2. Et les phrases aléatoires générées à l'Étape 2 l'ont été dans le but exprès de les fournir en entrée aux différents outils de lemmatisation existants, pour en quantifier la performance.

Il faut toutefois se rappeler que le but principal de ce projet de recherche était de créer un algorithme pour générer des phrases aléatoires automatiquement lemmatisées. L'application de ces textes pour l'évaluation d'outils de lemmatisation existants ne sert ici qu'à valider l'approche (« preuve de concept »). Cette évaluation d'outils existants sera donc limitée dans sa portée, avec l'objectif principal de démontrer l'utilité des textes aléatoires générés, et donc la pertinence de ce mémoire.

On décrit en premier lieu (Section 6.1) les quelques outils de lemmatisation soumis à l'évaluation. On discute ensuite de la projection des étiquettes morpho-syntaxiques pour chaque outil (Section 6.2). Cette projection a pour but de permettre une comparaison directe entre les résultats de ces outils et les étiquettes fournies au cours de ce projet pour les textes aléatoires. On introduit ensuite les textes auxquels les outils ont été confrontés (Section 6.3). Divers textes, en plus des textes aléatoires automatiquement lemmatisés, seront utilisés pour quantifier la performance de désambiguïsation à la Section 6.4. On conclut le chapitre avec les comparaisons globales de performance des différents outils, selon les textes imposés (Section 6.4.5).

6.1. Outils de lemmatisation existants à évaluer

De nombreux outils de lemmatisation ont été développés par différents groupes de chercheurs ou organismes, tel que discuté à la Section 2.1.1. Certains de ces outils sont demeurés au stade d'outils de recherche, disponibles ou non au grand public, d'autres ont atteint le statut de produit commercial, tandis qu'on retrouve aussi quelques outils disponibles gratuitement sur le web. Dans le contexte du projet de recherche actuel, seuls quatre outils ont été sélectionnés pour être soumis aux divers textes à lemmatiser. Une description de chacun est fournie plus bas, de même que la pertinence de leur sélection dans le contexte du projet actuel.

Mais considérant la portée limitée de l'évaluation d'outils existants et le fait que cette analyse ne serve qu'à valider la pertinence du générateur de textes aléatoires, l'emphase sera mise sur le premier outil décrit : « TreeTagger » ainsi que sur le lemmatiseur développé dans le cadre de ce projet. Des résultats détaillés seront fournis pour ces deux outils (Section 6.4), ce qui inclut une comparaison des classes grammaticales prédites pour tous les mots des textes aléatoires. Les autres outils seront tout de même présentés, mais seuls quelques résultats seront discutés.

6.1.1. Outil TreeTagger

L'outil TreeTagger, déjà discuté à la Section 2.1.1, tire son nom du fait qu'il est bâti à l'aide d'arbres de décision (« *decision trees* »), et que sa fonction unique est d'étiqueter (« *tag* » en anglais) chaque mot d'un texte en fonction de sa classe grammaticale. Cet outil, développé originalement par Schmid (1995, 2013), est disponible sous quelques formats différents. Tout d'abord, il en existe une version de base, exécutable à l'aide d'invites de commandes Windows (Figure 6.1). Une interface graphique a ensuite été mise au point par Ciarán Ó Duibhín

(https://www3.smo.uhi.ac.uk/oduibhin/oideasra/interfaces/winttinterface.htm). Une capture d'écran de cette interface est illustrée à la Figure 6.2. Finalement, une version en ligne a été développée par le Centre de traitement automatique du langage (CENTAL) de l'Université Catholique de Louvain en Belgique (Figure 6.3). Cette version en ligne, permet facilement, sans installation de logiciel, d'appliquer l'outil TreeTagger à des fichiers texte, ou à du texte tapé directement dans l'outil web. C'est cette version en ligne qui a été utilisée principalement ici.

L'outil TreeTagger est disponible pour plusieurs langues, dont le français. On en déduit que les mêmes algorithmes sont utilisés peu importe la langue, mais des banques de données lexicales et des règles de grammaire spécifiques à chaque langue ont été introduites au programme.

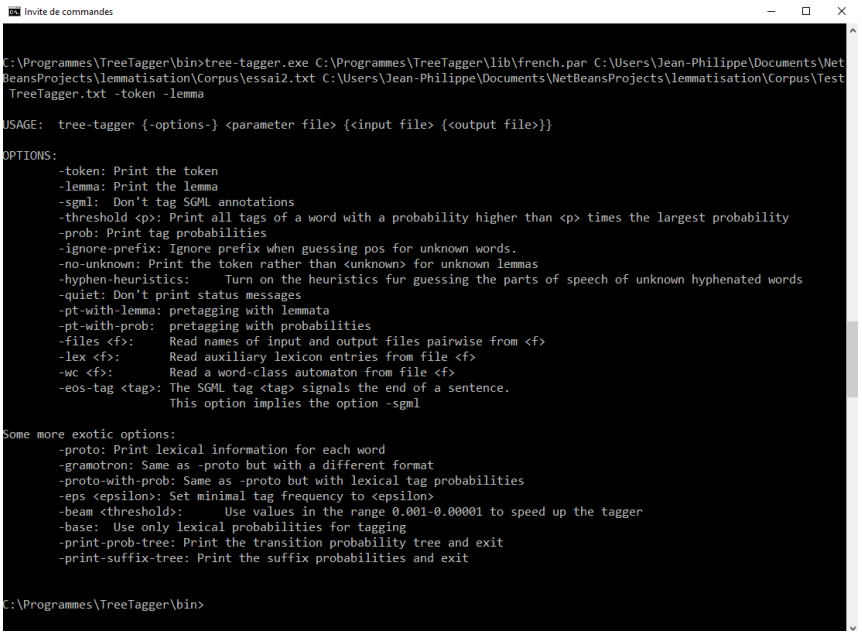


Figure 6.1 : Outil TreeTagger – Invite de commandes Windows

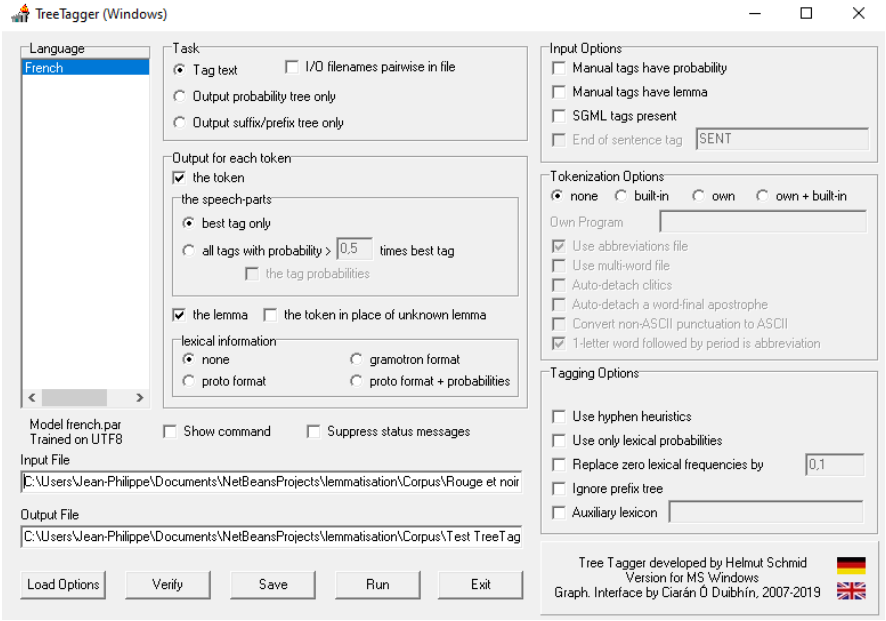


Figure 6.2 : Outil TreeTagger – Interface graphique

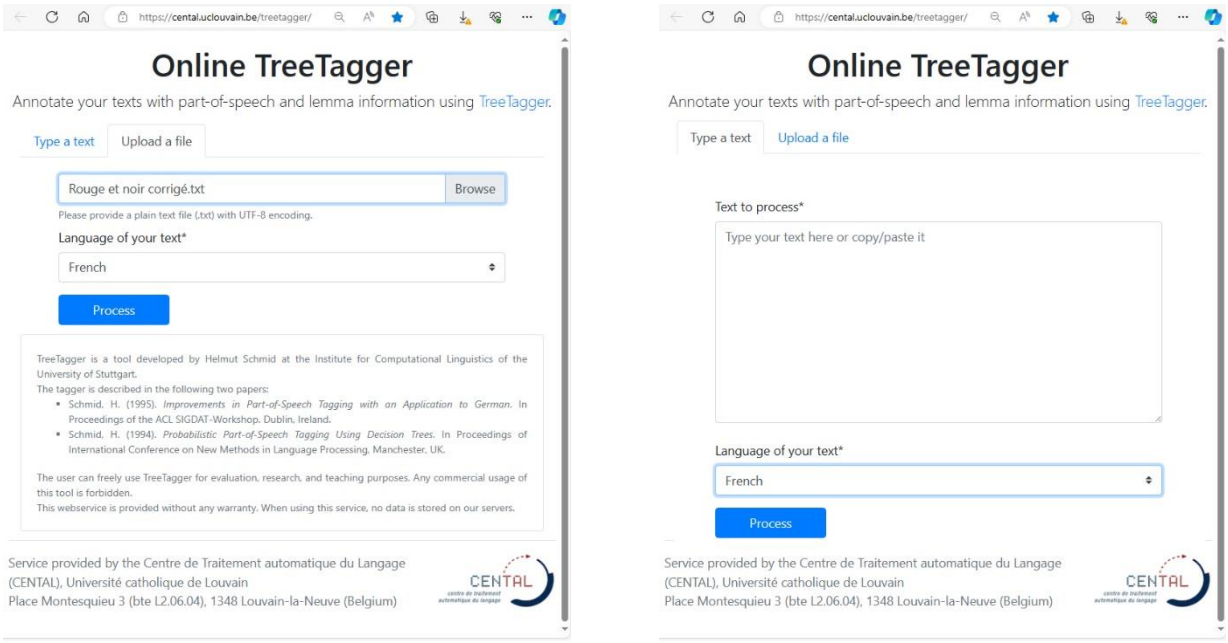
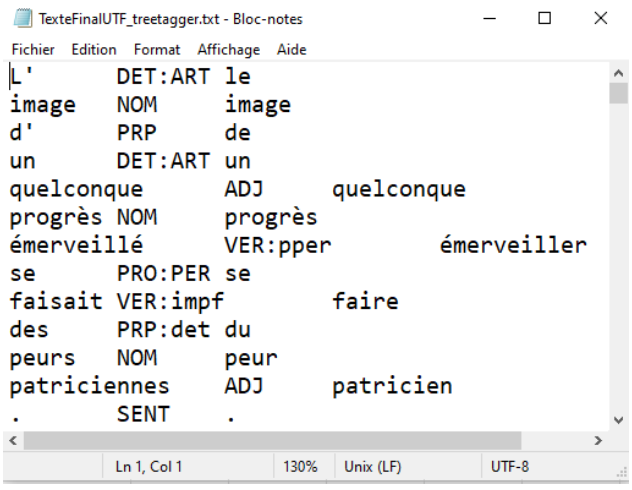


Figure 6.3 : Outil en ligne TreeTagger, (a) téléverser un fichier texte, (b) taper directement un texte à lemmatiser

L'outil TreeTagger (peu importe la version utilisée), accepte en entrée un fichier texte au format UTF-8, et fournit en sortie, pour chacun des mots du texte, son étiquette morpho-syntaxique (classe grammaticale et certains autres détails), ainsi que son lemme. Le fichier en sortie, qu'on peut ouvrir soit dans un outil de traitement de texte (« Bloc-notes », « Microsoft Word » ou autre) ou dans un chiffrier (« Excel »), est très simple à interpréter et à utiliser. Un exemple est fourni à la Figure 6.4, pour une phrase générée aléatoirement par l'algorithme bâti pour ce projet. L'utilisation d'un chiffrier rend le fichier plus facilement lisible, et favorise l'analyse ultérieure, qui impliquera la comparaison des résultats avec les lemmes et classes grammaticales automatiquement générées par l'algorithme de ce projet.



	A	B	C
1	L'	DET:ART	le
2	image	NOM	image
3	d'	PRP	de
4	un	DET:ART	un
5	quelconque	ADJ	quelconque
6	progrès	NOM	progrès
7	émerveillé	VER:pper	émerveiller
8	se	PRO:PER	se
9	faisait	VER:impf	faire
10	des	PRP:det	du
11	peurs	NOM	peur
12	patriciennes	ADJ	patricien
13	.	SENT	.

Figure 6.4 : Résultat TreeTagger pour une phrase type. (a) Fichier texte (b) Même fichier ouvert dans Excel

6.1.2. Outil Cordial

L'outil Cordial, tel que présenté à la Section 2.1.1, est considéré comme l'un des outils de lemmatisation les plus puissants pour le français (il n'est pas utilisé en d'autres langues). Il s'agit d'un outil commercial, vendu comme module d'extension (« *plug-in* ») disponible pour plusieurs logiciels, incluant Microsoft Word. Une fois le logiciel Cordial installé, on peut donc accéder à ses fonctionnalités directement à partir de Word, en activant son ruban au niveau de la barre de menu (Figure 6.5). Ce ruban donne ensuite accès à de nombreuses fonctions. En effet, Cordial est bien plus qu'un lemmatiseur, puisqu'il permet la correction en direct dans Word et l'analyse d'un texte sous toutes ses formes. Pour le besoin du projet actuel, c'est la fonction d'analyse de la phrase qui nous intéresse, puisque c'est cette option qui nous renseigne sur les lemmes correspondant à chaque mot de la phrase, ainsi que leurs classes grammaticales. La Figure 6.5 illustre le bouton donnant accès à cette fonction, qui s'applique alors sur la première phrase de toute portion de texte préalablement sélectionnée dans Word.

Une fois ce bouton appuyé, les résultats de l'analyse de la phrase s'affichent dans une nouvelle fenêtre (Figure 6.6). Dans cette fenêtre, on retrouve la liste de tous les mots de la phrase, chacun occupant une rangée distincte. Pour chacun de ces mots, Cordial fournit ensuite de l'information détaillée, résumée ici :

- Le mot (incluant la ponctuation – apostrophe, virgule, point)
- Le lemme (forme canonique)
- Le numéro de la proposition dont le mot fait partie
- Le type de proposition dont le mot fait partie (indépendante, subordonnée, etc.)
- La fonction de la proposition (sujet, verbe, attribut)
- Le groupe dont le mot fait partie (nominal, pronominal)
- Le type détaillé (e.g. pronom, nom) incluant genre et nombre, temps et personne de verbe
- De l'information sémantique (sens contextuel)

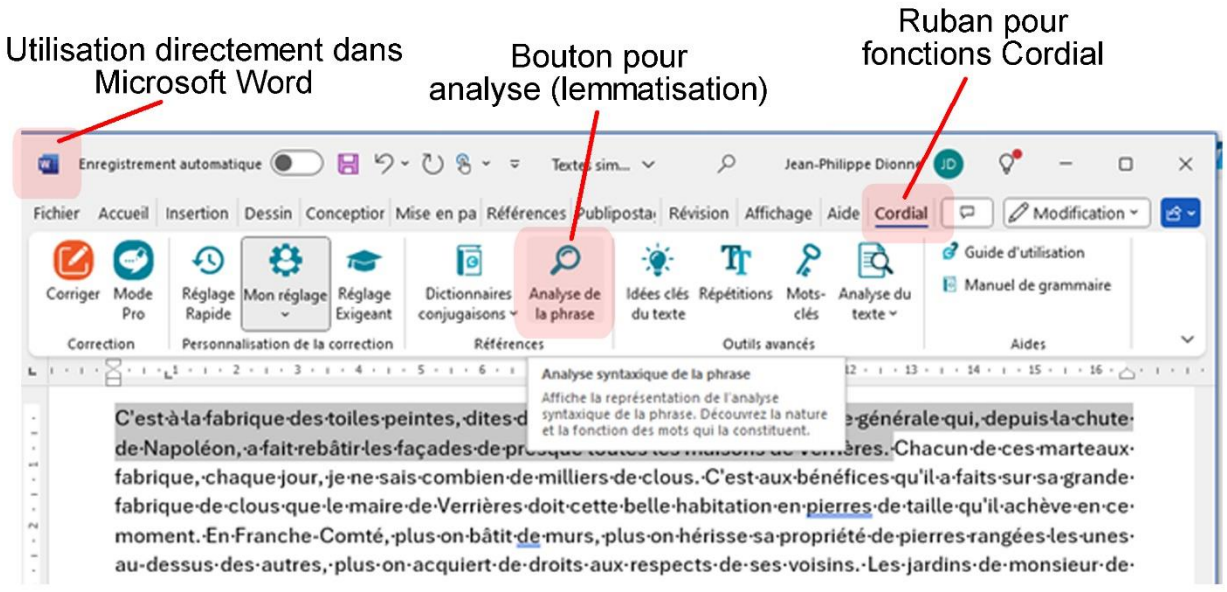
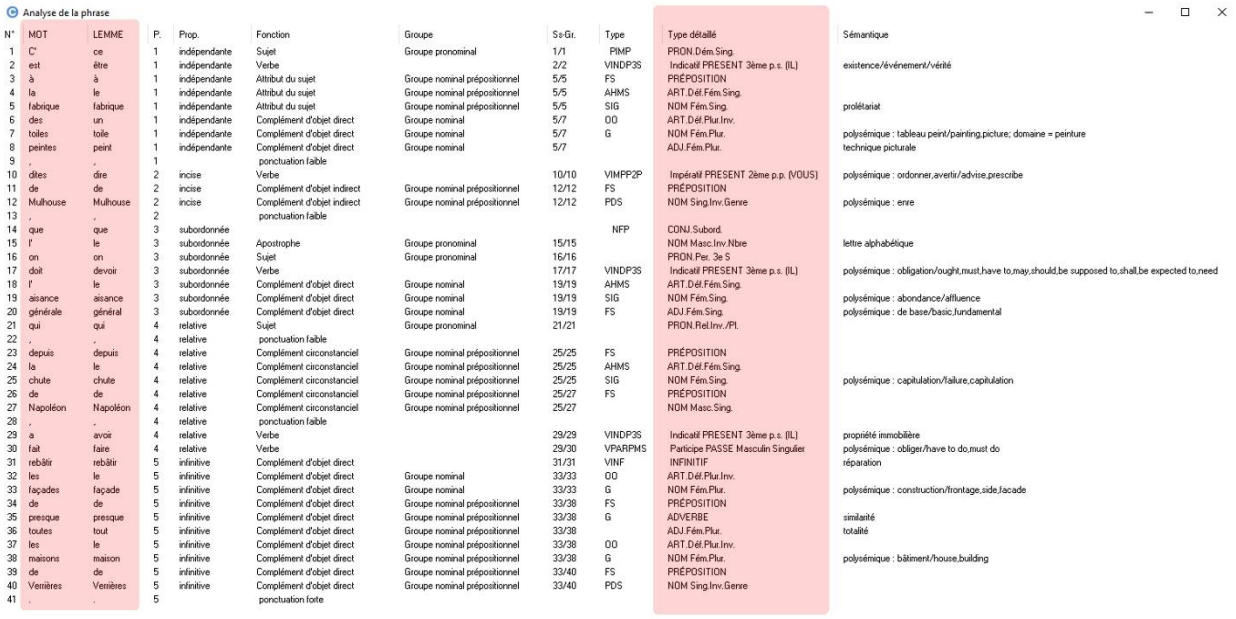


Figure 6.5 : Utilisation de Cordial directement dans Microsoft Word pour l'analyse des phrases



N°	MOT	LEMME	P.	Prop.	Fonction	Groupe	Ss-Gr.	Type	Type détaillé	Sémantique
1	C'	ce	1	indépendante	Sujet	Groupe pronominal	1/1	PIMP	PRON.Dém.Sing.	
2	est	être	1	indépendante	Verbe		2/2	VINDP3S	Indicatif PRESENT 3ème p.s. (IL)	existence/événement/vérité
3	à	à	1	indépendante	Attribut du sujet	Groupe nominal prépositionnel	5/5	FS	PRÉPOSITION	
4	la	le	1	indépendante	Attribut du sujet	Groupe nominal prépositionnel	5/5	AHMS	ART.Déf.Fém.Sing.	
5	fabrique	fabrique	1	indépendante	Attribut du sujet	Groupe nominal prépositionnel	5/5	SIG	NOM.Fém.Sing.	prolétariat
6	des	un	1	indépendante	Complément d'objet direct	Groupe nominal	5/7	OO	ART.Déf.Plur.Inv.	
7	toiles	toile	1	indépendante	Complément d'objet direct	Groupe nominal	5/7	G	NOM.Fém.Plur.	polysémique : tableau peint/painting,picture; domaine = peinture
8	peintes	peint	1	indépendante	Complément d'objet direct	Groupe nominal	5/7		ADJ.Fém.Plur.	technique picturale
9	,	,	1		ponctuation faible					
10	dites	dire	2	incise	Verbe		10/10	VIMPP2P	Impératif PRESENT 2ème p.p. (VOUS)	polysémique : ordonner,avertir/advise,prescribe
11	de	de	2	incise	Complément d'objet indirect	Groupe nominal prépositionnel	12/12	FS	PRÉPOSITION	
12	Mulhouse	Mulhouse	2	incise	Complément d'objet indirect	Groupe nominal prépositionnel	12/12	PDS	NOM Sing.Inv.Genre	polysémique : enre
13	,	,	2		ponctuation faible					
14	que	que	3	subordonnée				NFP	CONJ.Subord.	
15	l'	le	3	subordonnée	Apostrophe	Groupe pronominal	15/15		NOM Masc.Inv.Nbre	lettre alphabétique
16	on	on	3	subordonnée	Sujet	Groupe pronominal	16/16		PRON.Per. 3e S	
17	doit	devoir	3	subordonnée	Verbe		17/17	VINDP3S	Indicatif PRESENT 3ème p.s. (IL)	polysémique : obligation/ought,must,have to,may,should,be supposed to,shall,be expected to,need
18	l'	le	3	subordonnée	Complément d'objet direct	Groupe nominal	19/19	AHMS	ART.Déf.Fém.Sing.	
19	aisance	aisance	3	subordonnée	Complément d'objet direct	Groupe nominal	19/19	SIG	NOM.Fém.Sing.	polysémique : abondance/affluence
20	générale	général	3	subordonnée	Complément d'objet direct	Groupe nominal	19/19	FS	ADJ.Fém.Sing.	polysémique : de base/basic,fundamental
21	qui	qui	4	relative	Sujet	Groupe pronominal	21/21		PRON.Rel.Inv./Pl.	
22	,	,	4	relative	ponctuation faible					
23	depuis	depuis	4	relative	Complément circonstanciel	Groupe nominal prépositionnel	25/25	FS	PRÉPOSITION	
24	la	le	4	relative	Complément circonstanciel	Groupe nominal prépositionnel	25/25	AHMS	ART.Déf.Fém.Sing.	
25	chute	chute	4	relative	Complément circonstanciel	Groupe nominal prépositionnel	25/25	SIG	NOM.Fém.Sing.	polysémique : capitulation/failure,capitulation
26	de	de	4	relative	Complément circonstanciel	Groupe nominal prépositionnel	25/27	FS	PRÉPOSITION	
27	Napoléon	Napoléon	4	relative	Complément circonstanciel	Groupe nominal prépositionnel	25/27		NOM Masc.Sing.	
28	,	,	4	relative	ponctuation faible					
29	a	avoir	4	relative	Verbe		29/29	VINDP3S	Indicatif PRESENT 3ème p.s. (IL)	propriété immobilière
30	fait	faire	4	relative	Verbe		29/30	VPARPMS	Participe PASSE Masculin Singulier	polysémique : obliger,/have to do,must do
31	rebâtir	rebâtir	5	infinitive	Complément d'objet direct		31/31	VINF	INFINITIF	réparation
32	les	le	5	infinitive	Complément d'objet direct	Groupe nominal	33/33	OO	ART.Déf.Plur.Inv.	
33	façades	façade	5	infinitive	Complément d'objet direct	Groupe nominal	33/33	G	NOM.Fém.Plur.	polysémique : construction/frontage,side,facade
34	de	de	5	infinitive	Complément d'objet direct	Groupe nominal prépositionnel	33/38	FS	PRÉPOSITION	
35	presque	presque	5	infinitive	Complément d'objet direct	Groupe nominal prépositionnel	33/38	G	ADVERBE	similarité
36	toutes	tout	5	infinitive	Complément d'objet direct	Groupe nominal prépositionnel	33/38		ADJ.Fém.Plur.	totalité
37	les	le	5	infinitive	Complément d'objet direct	Groupe nominal prépositionnel	33/38	OO	ART.Déf.Plur.Inv.	
38	maisons	maison	5	infinitive	Complément d'objet direct	Groupe nominal prépositionnel	33/38	G	NOM.Fém.Plur.	polysémique : bâtiment/house,building
39	de	de	5	infinitive	Complément d'objet direct	Groupe nominal prépositionnel	33/40	FS	PRÉPOSITION	
40	Verrières	Verrières	5	infinitive	Complément d'objet direct	Groupe nominal prépositionnel	33/40	PDS	NOM Sing.Inv.Genre	
41	.	.	5		ponctuation forte					

Figure 6.6 : Information fournie par Cordial pour chaque phrase (affichage à l'écran). En rouge, l'information utilisée dans le contexte actuel (mot, lemme et type détaillé)

Cordial offre donc davantage d'information que ce dont on a besoin pour le projet actuel. À la Figure 6.6, les trois colonnes requises pour le projet sont celles surlignées en rouge. Les autres colonnes sont donc ignorées ici. On constate que ces trois colonnes en rouge fournissent une information équivalente à celle fournie par l'outil TreeTagger discuté à la section précédente.

Cordial offre aussi une analyse du texte dans son ensemble (plutôt que phrase par phrase). La Figure 6.7 donne un aperçu ce qui est fourni. On retrouve par exemple des statistiques sur le nombre total de mots du texte, le nombre total de phrases et le nombre moyen de mots par phrase (Figure 6.7a). Ces trois données sont pertinentes dans le cadre de ce projet. Mais Cordial fournit aussi de l'information sur le style de texte (e.g. juridique, scientifique, littéraire), qui est moins pertinente. Finalement, à la Figure 6.7b, Cordial fournit aussi des statistiques sur le nombre d'occurrences des divers lemmes du texte (par ordre de fréquences ou alphabétique). Cette information, bien que pertinente de façon générale dans le contexte de la lemmatisation, ne sera pas utilisée pour l'analyse limitée de ce chapitre.

Malheureusement, aucun de ces tableaux ne peut facilement être exporté ni copié dans un outil tiers (chiffrier ou traitement de texte). À toute fin pratique, ces tableaux sont donc comme des images. Ainsi, l'information pour chaque phrase, pour être analysée et comparée avec l'information obtenue à partir d'autres outils, doit être recopiée manuellement. De plus, en ce qui concerne l'analyse de la phrase, il faut procéder une phrase à la fois, ce qui rend pénible l'utilisation de cet outil commercial dans un contexte de recherche en linguistique. Ces contraintes expliquent que l'emphase ait été mise sur TreeTagger et non Cordial, pour démontrer la pertinence du générateur de textes aléatoires automatiquement lemmatisés.

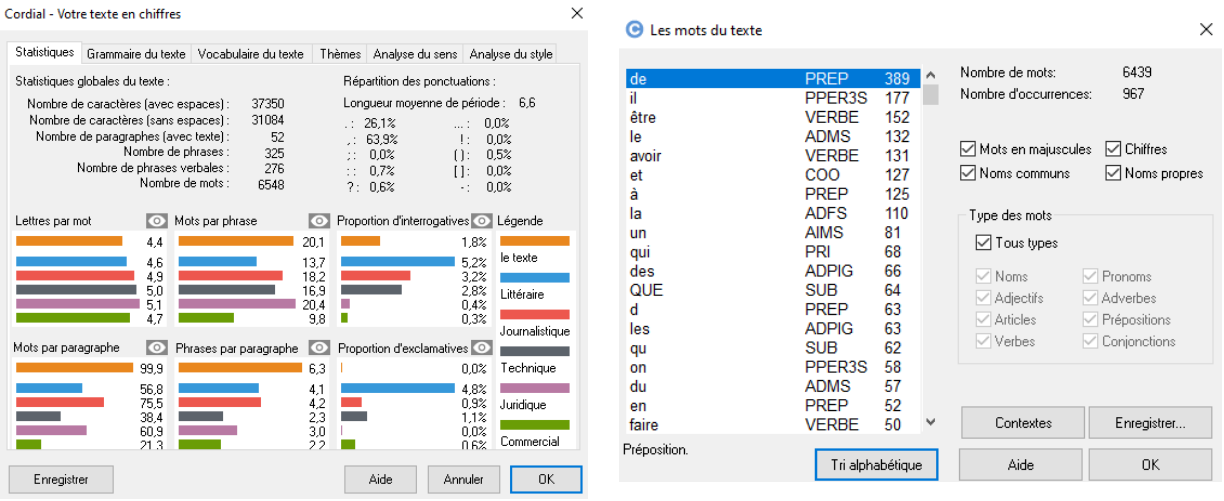


Figure 6.7 : Informations additionnelles fournies par Cordial pour chaque phrase (affichage à l'écran), incluant le style du texte et la fréquence des mots utilisés

6.1.3. Outil de base gratuit sur le web

D'autres outils de lemmatisation, souvent plus simples, sont disponibles sur le web. Un d'entre eux est présenté ici, soit celui développé par Jérôme Pasquelin, et que l'on peut retrouver à l'adresse https://www.jerome-pasquelin.fr/tools/outil_lemmatisation.php. Comme la page d'accueil web l'indique (Figure 6.8), l'outil sert à analyser et à trouver les lemmes d'un texte. L'outil compte aussi le nombre total de mots, et de mots uniques. L'utilisation de l'outil est simple : on tape directement ou on « colle » du texte à la fenêtre d'entrée, puis on appuie sur le bouton « OK » pour obtenir le résultat de l'analyse.

Ce que l'auteur appelle un « remplacement » est en fait le remplacement du mot original du texte par son lemme associé. Ainsi, par exemple, le mot « ferais » serait remplacé par son lemme « faire ». L'opération appelée ici « remplacement » est donc le résultat de la lemmatisation du texte.

La Figure 6.9 offre un exemple d'analyse d'un texte. Le texte entré ici est un ensemble de quelques phrases éparses du roman « Le Rouge et le Noir » (on discute de ces phrases à la Section 6.3.2). Un diagramme à barres fournit la fréquence d'utilisation des lemmes les plus utilisés dans le texte (et non les mots). Par exemple, on voit que le verbe « avoir » a été utilisé 6 fois dans le texte, selon cette analyse, et que le verbe « devoir » a quant à lui été utilisé 5 fois. Aussi inclus à la Figure 6.9 est un tableau montrant les différents mots fléchis associés aux différents lemmes répertoriés dans le texte. Par exemple, le verbe « être » a été retrouvé sous sa forme infinitive « être », de même que sous les formes « est » et « sont », tandis que le verbe « avoir » a été retrouvé sous les formes « a » et « avoir ». Le lemme « toile » a quant à lui été retrouvé sous les formes « toile » et « toiles ». On note aussi que 57 mots ont été ainsi remplacés par leur lemme dans ce texte. On note au passage une faute d'orthographe sur la page web : « *remplcaement* » qui témoigne du peu d'effort investi au niveau de la qualité pour ce site.

Cet outil ne fournit pas de résultat « mot par mot », ce qui limite son utilité dans le contexte de ce projet de mémoire. On n'en fera donc qu'une évaluation très limitée dans ce chapitre.

Figure 6.8 : Outil de lemmatisation simple retrouvé sur le web (Lemmatisation : outil pour regrouper les mots d'une même famille (jerome-pasquelin.fr)

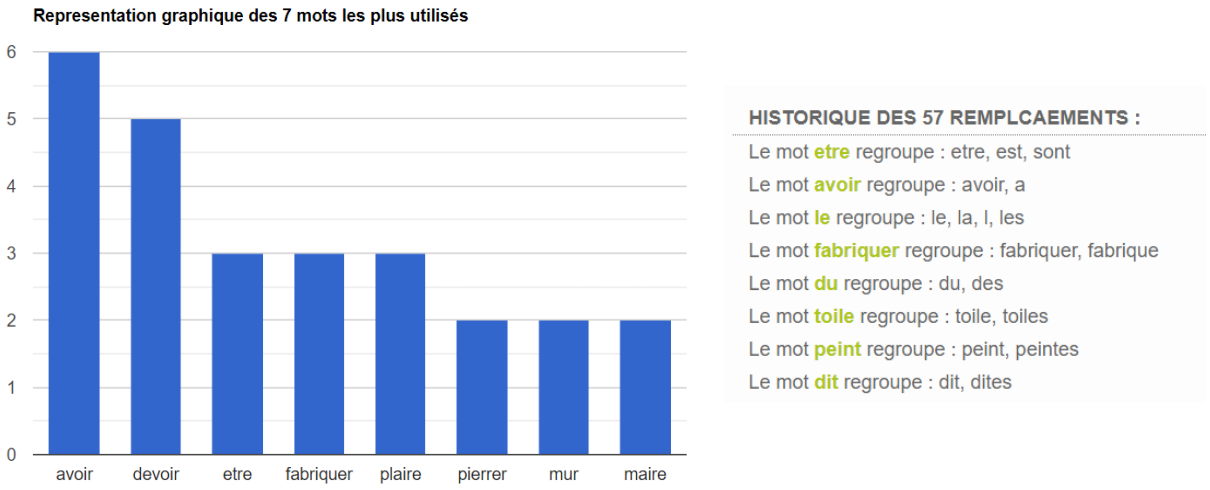


Figure 6.9 : Outil de lemmatisation simple. Exemple de résultats en sortie (capture d'écran)

6.1.4. Outil de lemmatisation développé pour ce projet

L'objectif principal de la création d'un outil de lemmatisation dans le cadre de ce projet était de bâtir une liste de lemmes et extraire quelques statistiques du corpus de référence pour aider à générer des phrases aléatoires automatiquement lemmatisées. Toujours est-il que l'outil développé s'est avéré efficace et qu'il est donc pertinent de comparer sa performance avec celles des autres outils discutés dans cette section. Il faut toutefois mentionner que cet outil part possiblement avec une longueur d'avance, puisque les textes lemmatisés au Chapitre 5 l'ont été à partir des banques de données générées par ce même outil. Malgré tout, sa performance de désambiguïsation des homographes dans le cas des textes aléatoires n'est pas pour autant garantie, car la structure des phrases bâties n'est pas la même que celles utilisées pour entrainer les algorithmes.

Il est inutile ici de décrire plus en détail l'outil de lemmatisation développé pour ce projet, considérant que c'est ce sur quoi les Chapitres 3 à 5 se sont penchés. Il suffit de mentionner que l'évaluation des textes du Chapitre 6, l'entrainement de l'algorithme sera basé sur le roman « Le Rouge et le Noir ». Il suffira ensuite d'effectuer l'évaluation en fournissant au programme chacun des textes décrits à la Section 6.3 (fichiers au format texte). L'information fournie en sortie a déjà été amplement décrite au Chapitre 5.

6.2. Projection des étiquettes grammaticales

L'évaluation des outils de lemmatisation existants implique qu'on compare leurs résultats à ceux de l'étalon (« *gold standard* ») représenté par les textes aléatoires automatiquement lemmatisés. Mais comme il n'existe pas de sets d'étiquettes standardisées pour décrire la classe grammaticale et la fonction des mots d'un texte, il a donc fallu s'assurer de transformer au besoin les étiquettes fournies par les outils de lemmatisation existants afin qu'elles s'arriment à celles fournies en sortie avec les textes automatiquement lemmatisés. Différentes approches d'étiquetage avaient été décrites aux Sections 2.1 et 3.1.1.

Deux niveaux d'étiquettes sont fournis en lien avec les textes automatiquement lemmatisés. Dans un premier lieu, on associe chaque mot à l'une des 9 classes grammaticales de base, qui avaient été décrites au Tableau 3.1 (verbe, adjectif, nom, adverbe, déterminant, pronom, préposition, conjonction et interjection). On s'assurera donc de bien effectuer la projection des classes grammaticales prédites par les différents outils sur les 9 classes du Tableau 3.1. Le niveau de performance des algorithmes sera ainsi d'abord quantifié sur la base de leur habileté à bien assigner chaque mot testé à l'une des 9 classes.

Un ensemble d'étiquettes plus précis peut aussi être utilisé, tel que décrit au Tableau 3.2. Les étiquettes de ce tableau ne seront cependant pas toutes utilisées. En particulier, aucune sous-catégorie d'adverbes ne sera assignée. Mais au-delà des étiquettes du Tableau 3.2, on peut comparer le genre et le nombre des noms et adjectifs, ainsi que le temps et la personne des verbes, puisque cette information est fournie en sortie de l'outil de génération de phrases aléatoires automatiquement lemmatisées. Cependant, on n'évaluera pas ici les outils de lemmatisation existants en fonction de ces étiquettes plus précises, vu la portée limitée de l'analyse. Davantage de détails sont fournis aux sections suivantes (6.2.1 à 6.2.4) concernant la projection des étiquettes en fonction des quatre outils soumis à l'évaluation.

6.2.1. Étiquettes pour l'outil TreeTagger

En sortie, l'outil TreeTagger assigne une étiquette à chaque mot du texte analysé, parmi un jeu de 33 étiquettes possibles. TreeTagger associe un court code à chacune de ces étiquettes. Ces codes sont fournis au Tableau 6.1, accompagnés de leur signification. Celle-ci est disponible en anglais seulement dans la documentation de TreeTagger, mais une traduction libre en français a été incluse à la colonne suivante.

La projection de ces étiquettes implique en premier lieu d'associer chacune d'elle à l'une des 9 classes grammaticales utilisées pour l'étalon (textes aléatoires automatiquement lemmatisés). Cette projection est affichée à l'avant-dernière colonne du Tableau 6.1, où on retrouve les chiffres 0 à 9, où le chiffre 0 implique une absence de classification, et les chiffres 1 à 9 correspondent aux classes grammaticales du Tableau 3.1. On note en premier lieu que pour les étiquettes TreeTagger « abréviation » et « symbole », aucune projection n'est possible sur les 9 étiquettes de base développées pour ce projet. Ceci ne pose pas problème dans le contexte actuel, puisque les textes à évaluer (décrits à la Section 6.3), ne comprennent ni abréviations ni symboles. On note aussi qu'aucune projection n'est possible pour les catégories « ponctuation », « ponctuation (citation) » et « fin de phrase ». C'est que l'outil développé pour ce projet ne compile pas de données concernant la ponctuation. Tout élément classifié par TreeTagger correspondant à l'une de ces trois catégories sera donc tout simplement exclu de l'analyse. Sinon, exception faite des cas associés aux codes « 0 » et « n/a », chacune des étiquettes de TreeTagger a été associée (projection) à l'une des 9 étiquettes de base développées pour ce projet.

La dernière colonne du Tableau 6.1 correspond aux étiquettes plus précises mentionnées à la Section 6.2, sur lesquelles les étiquettes de TreeTagger ont été projetées. Deux types de noms sont inclus, soit les noms communs (31) et les noms propres (32). Au niveau des déterminants, trois sous-catégories sont représentées, soit les articles (51), les déterminants possessifs (53) et les déterminants numéraux (54). Pour ce qui est des pronoms, TreeTagger inclut les pronoms personnels (61), démonstratifs (62), possessifs (63), indéfinis (64), relatifs (65), ainsi qu'une classification plus générale (6). Deux types de prépositions sont considérées, soit les prépositions usuelles (7), ainsi que les « prépositions plus article » (71), qui impliquent une contraction (à+le = au, à+les = aux, de+le = du, de+les = des).

Les verbes quant à eux, sont étiquetés en fonction des temps impliqués. Au chiffre « 1 » caractérisant les verbes, on ajoute à l'étiquette un code pour le temps de verbe, à partir des codes de temps de verbes apparaissant au Tableau 3.4. Les temps de verbes considérés par TreeTagger sont l'infinitif (10), le présent (11), l'imparfait (12), le passé simple (13), le futur (14), le subjonctif présent (15), le subjonctif imparfait (16), l'impératif (17), le conditionnel (18), le participe présent (19) et le participe passé (110).

Aucune subdivision d'étiquette additionnelle n'est effectuée pour les adjectifs (2), les adverbes (4), les conjonctions (8) et les interjections (9). Il est à noter que TreeTagger ne permet pas de distinguer les personnes de verbe, ni le genre (féminin ou masculin) et le nombre (singulier ou pluriel) des adjectifs, noms, pronoms et déterminants. Il est aussi à noter que la documentation de TreeTagger fait erronément référence au mot « *pronoun* » (Tableau 6.1) pour décrire les déterminants possessifs (« ma », « ta », « sa », etc.). Mais cette erreur n'affecte en rien la performance de l'outil.

Une méthode Java simple (non détaillée ici) a été mise au point pour procéder à la projection, c'est-à-dire à assigner, pour chaque mot lemmatisé par TreeTagger, une classe grammaticale (chiffre de 0 à 9) et une étiquette plus détaillée (dernière colonne du Tableau 6.1).

Tableau 6.1 : Projection des étiquettes de TreeTagger

	Signification		Projection	
Code Tree Tagger	Documentation originale de TreeTagger (anglais)	Traduction libre (français)	Classe	Détail
ABR	abbreviation	abréviation	0	0
ADJ	adjective	adjectif	2	2
ADV	adverb	adverbe	4	4
DET:ART	article	article	5	51
DET:POS	possessive pronoun (ma, ta, ...)	déterminant possessif	5	53
INT	interjection	interjection	9	9
KON	conjunction	conjonction	8	8
NAM	proper name	nom propre	3	32
NOM	noun	nom commun	3	31
NUM	numeral	numéral	5	54
PRO	pronoun	pronom	6	6
PRO:DEM	demonstrative pronoun	pronom démonstratif	6	62
PRO:IND	indefinite pronoun	pronom indéfini	6	64
PRO:PER	personal pronoun	pronom personnel	6	61
PRO:POS	possessive pronoun (mien, tien, ...)	pronom possessif	6	63
PRO:REL	relative pronoun	pronom relatif	6	65
PRP	preposition	préposition	7	7
PRP:det	preposition plus article (au,du,aux,des)	préposition avec article	7	71
PUN	punctuation	ponctuation	n/a	n/a
PUN:cit	punctuation citation	ponctuation (citation)	n/a	n/a
SENT	sentence tag	fin de phrase	n/a	n/a
SYM	symbol	symbole	0	0
VER:cond	verb conditional	verbe - conditionnel	1	18
VER:futu	verb futur	verbe - futur	1	14
VER:impe	verb imperative	verbe - impératif	1	17
VER:impf	verb imperfect	verbe - imparfait	1	12
VER:infi	verb infinitive	verbe - infinitif	1	10
VER:pper	verb past participle	verbe - participe passé	1	110
VER:ppre	verb present participle	verbe - participe présent	1	19
VER:pres	verb present	verbe - présent	1	11
VER:simp	verb simple past	verbe - passé simple	1	13
VER:subi	verb subjunctive imperfect	verbe - subjonctif imparfait	1	16
VER:subp	verb subjunctive present	verbe - subjonctif présent	1	15

6.2.2. Étiquettes pour l'outil Cordial

Tandis que l'outil TreeTagger ne dispose que de 33 étiquettes distinctes pour classifier les mots, l'outil Cordial offre un étiquetage beaucoup plus détaillé. Par exemple, alors que TreeTagger, pour les verbes, ne faisait qu'en spécifier le temps, Cordial précise aussi la personne (1ère, 2e, 3e personnes du singulier ou du pluriel). Pour les noms communs et les adjectifs, Cordial précise aussi le genre (masculin, féminin), et le nombre (singulier, pluriel), ce que ne faisait pas non plus TreeTagger. Même chose pour les déterminants, le cas échéant. Les adverbes sont aussi groupés en plusieurs catégories différentes (manière, quantité, temps, lieu, affirmation, négation et doute). Ces regroupements d'adverbe diffèrent de ceux suggérés au Tableau 3.2, qui étaient au nombre de quatre (intensité, caractérisation, quantification et spatio-temporel). Pour ce qui est des pronoms, Cordial dénombre aussi les pronoms possessifs, personnels, relatifs, indéfinis, interrogatifs et exclamatifs. Les pronoms peuvent avoir dans plusieurs cas un genre et un nombre, et parfois une personne (dans le cas possessif). Ces étiquettes sont semblables à celles du Tableau 3.2.

Pour ce qui est des adjectifs qualificatifs, comme « grande », « beau », etc., Cordial fournit aussi le genre et le nombre, ce que ne faisait pas TreeTagger. Cependant, Cordial considère comme adjectifs des mots qui sont considérés dans le présent projet comme des déterminants. Par exemple, dans le segment de phrase « *cette* maison est bleue », le mot « cette » est considéré comme un déterminant démonstratif dans ce projet (code 52), tandis que l'outil Cordial classifie ce mot comme un « *adjectif* démonstratif ». Au moment de la projection des étiquettes de Cordial, de tels « adjectifs démonstratifs » seront toutefois classifiés (projection) comme étant des « déterminants » (classe grammaticale 5) pour les besoins de la comparaison. La même observation s'applique pour ce que Cordial qualifie d'adjectif possessif (« *ma* maison ») et adjectif indéfini (« *certaines* maisons sont vertes »). Là encore, la projection fera en sorte que ces adjectifs (selon Cordial) seront plutôt considérés comme déterminants, dans le cadre de ce projet. Ainsi, si Cordial détermine que le mot « cette » est un adjectif et que dans le projet actuel on détermine qu'il s'agit plutôt d'un déterminant, on ne considèrera pas que Cordial a erré.

La sélection des noms pour les étiquettes, qui de toute évidence n'est pas standardisée, est hors de la portée du projet actuel. Aucun commentaire éditorial ne serait fait ici pour juger du nom d'étiquette qui serait le plus approprié. Nous laissons ce débat aux linguistes et grammairiens. Nous nous contentons ici de nous assurer une bonne « projection » entre les termes utilisés par chaque outil et ceux choisis pour ce travail.

Mais malheureusement, tel que mentionné plus tôt, l'outil Cordial, du moins dans sa version commerciale utilisée ici, ne fournit pas les résultats de son étiquetage dans un format se prêtant à l'analyse de longs textes. En effet, les données en sortie ne peuvent pas être copiées ou transférées dans d'autres outils, autrement que de façon manuelle (recopier un à un chaque code). Pour cette raison, et donc pour simplifier et limiter l'effort, l'analyse syntaxique avec l'outil Cordial se limitera ici à déterminer uniquement la classe grammaticale (chiffre de 1 à 9), selon les définitions du Tableau 3.1. L'étiquetage plus précis, incluant la détermination du genre et du nombre des noms, de même que du temps et des personnes de verbes, ne sera qu'effleuré.

Puisque le transfert des résultats de Cordial se fait complètement manuellement, aucune méthode Java n'a été mise au point pour la projection des données, comme on l'a fait pour l'outil TreeTagger. L'évaluation de l'outil Cordial sera donc beaucoup plus sommaire que celle effectuée pour TreeTagger.

6.2.3. Étiquettes pour l'outil de base gratuit sur le web

L'outil développé par Jérôme Pasquelin, contrairement aux outils TreeTagger et Cordial, ne fournit aucun tableau en sortie listant le lemme et l'étiquette de chaque mot. Comme on l'a vu à la Figure 6.9, cet outil ne fournit que des résultats globaux concernant le texte analysé. L'outil effectue forcément une analyse de chaque mot individuel pour son analyse, mais seuls les résultats globaux sont affichés en sortie. De plus, toujours à la Figure 6.9, on constate que cet outil ne fournit de l'information que sur les lemmes, et non sur les classes grammaticales. Par exemple, les trois premiers mots apparaissant sur le diagramme à barres (« avoir », « devoir » et « être ») sont des homographes, et il n'est nullement spécifié s'il s'agit ici des formes verbales ou nominales de ces termes. Aucun étiquetage n'a donc été effectué par cet outil, si bien qu'aucune projection n'est nécessaire.

Et comme aucun tableau ne fournit les résultats individuels pour chaque mot du texte, l'évaluation de cet outil sera encore plus sommaire que celle effectuée pour Cordial. Dans ce cas-ci, il faudra y aller d'inférence pour l'évaluation. Plus précisément, on se servira des résultats fournis dans le sommaire pour déduire certaines informations concernant les mots individuels, et ainsi pouvoir commenter sur l'efficacité de cet outil.

6.2.4. Étiquettes pour l'outil de lemmatisation développé pour ce projet

Puisque l'outil de lemmatisation développé pour ce projet a été bâti parallèlement au générateur de textes aléatoires automatiquement lemmatisés devant servir d'étalon, et par le même programmeur, on devrait s'attendre à ne *pas* avoir à effectuer de projection. En effet, les mêmes codes, issus des Tableaux 3.1 et 3.2 ont été utilisés dans les deux cas. Cependant, il faut mentionner qu'aucun effort n'a été fait au niveau du lemmatiseur pour désambiguïser les formes verbales pouvant appartenir à plusieurs temps ou plusieurs personnes. La forme verbale « aime » par exemple, tel que mentionné précédemment, se réfère en effet à trois possibilités différentes de temps de verbe (présent, subjonctif et impératif) et trois personnes différentes (1ère, 2e et 3e personnes du singulier). Cet outil de lemmatisation n'effectue donc pas une lemmatisation complète, car on ne peut, dans tous les cas, préciser le temps et la personne de chaque verbe. De plus, cet outil ne peut non plus déterminer si le nom « fils » est au pluriel ou au singulier. En effet, on ne s'est penché dans ce projet que sur les homographes appartenant à des classes grammaticales différentes.

Pour cette raison, on se contentera ici, pour l'évaluation de l'outil de lemmatisation développé pour ce projet, de ne comparer que les classes grammaticales (codes de 1 à 9), et non les étiquettes détaillées. Là était d'ailleurs l'objectif principal de cet outil, pour lequel l'apprentissage machine a été déployé : déterminer la classe grammaticale de chaque mot.

Aucune projection n'est donc nécessaire, puisque les mêmes codes ont été utilisés (ceux du Tableau 3.1) pour définir les classes grammaticales, pour l'outil de lemmatisation et pour les textes aléatoires automatiquement lemmatisés.

6.3. Textes auxquels soumettre les lemmatiseurs

Le but premier de ce projet de recherche est de générer des textes aléatoires automatiquement lemmatisés. Ces textes peuvent ensuite être soumis à des outils de lemmatisation pour en évaluer l'efficacité. Il va donc de soi que les outils de lemmatisation existants listés à la Section 6.1 seront testés sur la base de ces textes aléatoires. Toutefois, les outils de lemmatisation seront aussi soumis à quelques autres textes, dans un but d'exploration ou pour investiguer quelques caractéristiques particulières. Le but de la présente section est de fournir et décrire tous les textes auxquels seront donc soumis les outils de lemmatisation.

6.3.1. Courts textes avec objectifs précis

Tel que mentionné à la Section 2.2, le plus grand défi d'un lemmatiseur est de désambiguïser les homographes. Pour mettre ce défi en relief, chaque lemmatiseur sera en premier lieu soumis à deux courts textes ne contenant quelques phrases.

Tous les mots du premier de ces textes (Figure 6.10) ont été sélectionnés avec soin afin qu'il ne contienne aucun homographe. On s'attend à ce que la performance des outils de lemmatisation soit maximale dans un tel cas, puisque aucune désambiguïsation n'est nécessaire. Il faut toutefois préciser que les quatre phrases de ce texte, affiché à la Figure 6.10, n'ont aucun lien entre elles.

```
Un camion du magasin, dont une conductrice habile surveillait trop
attentivement sa vitesse, filait très bruyamment autour du quartier
très laid, et ne ralentissait guère. Puisque ma mère patientait depuis
trois jours, elle voulut enfin démontrer aux gens cette impatience
qui caractérise chaque actionnaire du clan auquel elle disait
appartenir. Malheureusement, elle ne put trouver une option assez
convenable pour celui dont elle voulait se distancer. Quiconque ferait
siens ces immenses bolides, découvrirait avec plaisir chez eux cet
aspect affreux qui répugnerait ainsi une femme sans scrupules.
```

Figure 6.10 : Court texte ne comprenant aucun homographe

En revanche, tous les mots du deuxième texte (Figure 6.11), sans exception, sont des homographes. On s'attend ici à ce que la performance des outils de lemmatisation soit plus faible, vu la difficulté associée à la désambiguïsation simultanée de tous ces homographes. En particulier, avec tous les mots étant des homographes, il est d'autant plus difficile de caractériser avec précision l'entourage d'un mot, autrement dit de déterminer les classes grammaticales des mots autour de chaque mot à évaluer. On a tenté d'inclure dans ce texte certains des homographes les plus courants des deux corpus de référence utilisés dans ce mémoire. On précise que les cinq phrases de la Figure 6.11 n'ont aucun lien entre elles, et n'ont aucun sens non plus.

```
Le vieux domestique fou dit que la porte, que le paysan contrôle, est
tout droit sortie de la table de dîner. Le commandant demande que le
bien fort courant de la chute demeure pas plus de neuf minutes si
haut. Rose est vite allée de nouveau dire qu'être dessous le mort nuit
plus que d'être assis entre les livres. Dites-leur que leur mépris
fait sourire tout libéral que porte ton complet. La voyant partie vers
le bas, l'autre personne reste étendue de tout son long.
```

Figure 6.11 : Court texte ne comprenant que des homographes

Un autre défi particulier pour les lemmatiseurs est de désambiguïser les homographes de même classe grammaticale, puisque ceux-ci ne peuvent généralement pas être distingués sur la base d'indices syntaxiques, tel qu'on le mentionnait à la Section 3.2.4. Le troisième court texte (Figure 6.12) inclut donc un certain nombre de phrases impliquant l'homographe « suis » (du verbe « être » ou du verbe « suivre »), l'homographe « fils » (nom au singulier ou au pluriel) et finalement l'homographe « convient » (du verbe « convenir » ou du verbe « convier »). L'analyse de la performance des lemmatiseurs concernant ce court texte se concentrera uniquement sur ces quelques homographes. Dans certains cas, il est impossible de procéder à la désambiguïsation en l'absence de contexte. Dans d'autres cas, le contexte est fourni, mais la désambiguïsation demande l'analyse de la sémantique de la phrase.

Je **suis** ton père. Tu es mon **fils**, donc je **suis** ton père. Comme il est parti le premier, je **suis** ton père. Tu me **suis** et je **suis** ton père. Les **fils** sont entremêlés. Les **fils** de la voisine sont venus l'aider. Le **fils** de la voisine est électricien. Les **fils** de la voisine sont électriciens. Comme **fils**, il ne se fait pas mieux, dit sa mère. Comme **fils**, j'ai utilisé les plus résistants pour brancher mon appareil. J'ai perdu mes **fils**. Cela me **convient**. Ils me **convient** à une rencontre.

Figure 6.12 : Court texte incluant certains homographes appartenant à la même classe grammaticale (« fils », « suis » et « convient »). Ces homographes sont surlignés en rouge

Finalement, le quatrième court texte (Figure 6.13) inclut deux segments de phrases offrant deux possibilités chacun pour le sens global. Pour le premier segment (« le boucher sale la coupe »), il peut s'agir d'un boucher qui n'est pas propre (« sale ») et qui coupe quelque chose dont le genre est féminin. Au contraire, il peut s'agir d'un boucher qui applique du sel (« sale ») à une coupe de viande. Il y a donc deux combinaisons possibles pour les homographes « sale », « la » et « coupe » : adjectif, pronom et verbe, ou encore verbe, déterminant et nom.

Pour le deuxième segment (« la belle ferme le voile »), il peut s'agir d'une « belle » (sous-entendu, une « belle femme »), qui ferme un voile. Ou il peut s'agir plutôt d'une belle ferme (« bâtiment agricole ») qui obstrue la vue à un individu de genre masculin. Il y a donc deux combinaisons possibles pour les homographes « belle », « ferme », « le » et « voile » : nom, verbe, déterminant et nom, ou encore adjectif, nom, pronom et verbe. Ces deux segments de phrases apparaissent soit seuls, ou accompagnés d'un autre segment offrant des indices sémantiques permettant, en théorie, de correctement désambiguïser la phrase, du moins, manuellement.

Le boucher sale la coupe. La pièce de viande, le boucher sale la coupe. Maintenant qu'il l'a préparée, le boucher sale la coupe. La belle ferme le voile. N'appréciant pas qu'il soit ouvert, la belle ferme le voile. Il veut admirer le paysage, mais la belle ferme le voile.

Figure 6.13 : Court texte de quelques phrases impliquant deux segments de phrase ambigus (« le boucher sale la coupe », et « la belle ferme le voile »)

6.3.2. Extraits du roman « Le Rouge et le Noir »

Le grand avantage des textes aléatoires automatiquement lemmatisés générés dans le cadre de ce mémoire, est qu'ils permettent l'évaluation d'outils de lemmatisation sans avoir à recourir à une lemmatisation manuelle, une tâche ardue. Mais on se rappelle que pour ce travail, des extraits substantiels de deux corpus de référence ont été lemmatisés manuellement pour l'entrainement manuel d'algorithmes d'apprentissage machine. Ceux-ci sont donc disponibles aussi pour l'évaluation des outils de lemmatisation évalués dans ce chapitre. Mais comme l'emphase demeure sur les textes aléatoires automatiquement lemmatisés (Section 6.3.3), seules quelques phrases du roman « Le Rouge et le Noir » serviront à l'évaluation. Un premier texte (Figure 6.14) contient des phrases éparses de ce roman dont tous les homographes sans exception ont été désambiguïsés *avec succès* par l'algorithme présenté au Chapitre 4 (donc un taux de succès de 100%). Les homographes y sont surlignés en rouge.

Un deuxième texte regroupe quant à lui des phrases éparses de ce même roman, mais qui n'ont pas aussi bien été désambiguïsées par l'algorithme décrit au Chapitre 4 (Figure 6.15). En effet, la performance de désambiguïsation des homographes pour ces phrases n'a été que de 80%. Les homographes sont de nouveau surlignés en rouge, et les homographes incorrectement désambiguïsés par l'algorithme développé ici sont soulignés. Ces deux textes offrent donc deux « cas limite » en ce qui a trait au niveau de difficulté pour la lemmatisation, en supposant qu'une phrase difficile à lemmatiser pour un outil en particulier l'est tout autant pour un autre outil. Il est à noter que l'outil web de Jérôme Pasquelin ne sera pas soumis à ces deux textes.

La petite ville de Verrières peut passer pour l'une des plus jolies de la Franche-Comté. Ses maisons blanches avec leurs toits pointus de tuiles rouges s'étendent sur la pente d'une colline, dont des touffes de vigoureux châtaigniers marquent les moindres sinuosités. Verrières est abrité du côté du nord par une haute montagne, c'est une des branches du Jura. Les cimes brisées du verra se couvrent de neige dès les premiers froids d'octobre. Ce ne sont pas cependant les scies à bois qui ont enrichi cette petite ville. Vingt marteaux pesants, et retombant avec un bruit qui fait trembler le pavé, sont élevés par une roue que l'eau du torrent fait mouvoir. Ce travail, si rude en apparence, est un de ceux qui étonnent le plus le voyageur qui pénètre pour la première fois dans les montagnes qui séparent la France de l'Helvétie. Si, en entrant à Verrières, le voyageur demande à qui appartient cette belle fabrique de clous qui assourdit les gens qui montent la grande rue, on lui répond avec un accent traînard. Elle est à monsieur le maire. A son aspect, tous les chapeaux se lèvent rapidement. Ses cheveux sont grisonnants, et il est vêtu de gris. Il est chevalier de plusieurs ordres, il a un grand front, un nez aquilin, et au total sa figure ne manque pas d'une certaine régularité. On trouve même, au premier aspect, qu'elle réunit à la dignité du maire de village cette sorte d'agrément qui peut encore se rencontrer avec quarante-huit ou cinquante ans. Mais bientôt le voyageur parisien est choqué d'un certain air de contentement de soi et de suffisance mêlé à je ne sais quoi de borné et de peu inventif. On sent enfin que le talent de cet homme-là se borne à se faire payer bien exactement ce qu'on lui doit, et à payer lui-même le plus tard possible quand il doit. Tel est le maire de Verrières, monsieur de Rênal. Après avoir traversé la rue d'un pas grave, il entre à la mairie et disparaît aux yeux du voyageur. Au-delà c'est une ligne d'horizon formée par les collines de la Bourgogne, et qui semble faite à souhait pour le plaisir des yeux. On lui apprend que cette maison appartient à monsieur de Rênal. Sa famille, dit-on, est espagnole, antique, et, à ce qu'on prétend, établie dans le pays bien avant la conquête de Louis XIV. Depuis, il rougit d'être industriel. L'a fait maire de Verrières. Ne vous attendez point à trouver en France ces jardins pittoresques qui entourent les villes manufacturières de l'Allemagne, Leipsick, Francfort, Nuremberg, etc. Quant au ruisseau public qui faisait aller la scie, monsieur de Rênal, au moyen du crédit dont il jouit à Paris, a obtenu qu'il fût détourné. Il est vrai que cet arrangement a été critiqué par les bonnes têtes de l'endroit. Ce sourire a porté un jour fatal dans l'âme de monsieur le maire, il pense depuis lors qu'il eût pu obtenir l'échange à meilleur marché. Une telle innovation vaudrait à l'imprudent bâtisseur une éternelle réputation de mauvaise tête, et il serait à jamais perdu auprès des gens sages et modérés qui distribuent la considération en Franche-Comté. La tyrannie de l'opinion, et quelle opinion. Le respect des sots, l'ébahissement des enfants, l'envie des riches, le mépris du sage. Heureusement pour la réputation de monsieur de Rênal comme administrateur, un immense mur de soutènement était nécessaire à la promenade publique qui longe la colline à une centaine de pieds au-dessus du cours du Doubs. Elle doit à cette admirable position une des vues les plus pittoresques de France. Mais, à chaque printemps, les eaux de pluie sillonnaient la promenade, y creusaient des ravins et la rendaient impraticable. Cet inconvénient, senti par tous, mit monsieur de Rênal dans l'heureuse nécessité d'immortaliser son administration par un mur de vingt pieds de hauteur et de trente ou quarante toises de long. Le parapet de ce mur pour lequel monsieur de Rênal a dû faire trois voyages à Paris, car l'avant-dernier ministre de l'Intérieur s'était déclaré l'ennemi mortel de la promenade de Verrières, le parapet de ce mur s'élève maintenant de quatre pieds au-dessus du sol. Et, comme pour braver tous les ministres présents et passés, on le garnit en ce moment avec des dalles de pierre de taille.

Figure 6.14 : Phrases éparses du roman « Le Rouge et le Noir » dont tous les homographes ont été désambiguïsés avec succès par l'algorithme décrit au Chapitre 4. Les homographes sont surlignés en rouge

C'est à la fabrique des toiles peintes, dites de Mulhouse, que l'on doit l'aisance générale qui, depuis la chute de Napoléon, a fait rebâtir les façades de presque toutes les maisons de Verrières. Chacun de ces marteaux fabrique, chaque jour, je ne sais combien de milliers de clous. C'est aux bénéfices qu'il a faits sur sa grande fabrique de clous que le maire de Verrières doit cette belle habitation en pierres de taille qu'il achève en ce moment. En Franche-Comté, plus on bâtit de murs, plus on hérisse sa propriété de pierres rangées les unes au-dessus des autres, plus on acquiert de droits aux respects de ses voisins. Les jardins de monsieur de Rênal, remplis de murs, sont encore admirés parce qu'il a acheté, au poids de l'or, certains petits morceaux de terrain qu'ils occupent. Malgré sa fierté, monsieur le maire a dû faire bien des démarches auprès du vieux Sorel, paysan dur et entêté, il a dû lui compter de beaux louis d'or pour obtenir qu'il transportât son usine ailleurs. Et, quoique cette position fût beaucoup plus avantageuse pour son commerce de planches de sapin, le père Sorel, comme on l'appelle depuis qu'il est riche, a eu le secret d'obtenir de l'impatience et de la manie de propriétaire, qui animait son voisin, une somme de francs. Une fois, c'était un jour de dimanche, il y a quatre ans de cela, monsieur de Rênal, revenant de l'église en costume de maire, vit de loin le vieux Sorel, entouré de ses trois fils, sourire en le regardant. Reprit vivement le geôlier, vous, monsieur le curé, on sait que vous avez livres de rente, du bon bien au soleil. On sait qu'il y a quarante-huit ans, j'ai hérité d'un champ qui rapporte livres. Cet arrangement convient de plus d'une façon, continua monsieur de Rênal, en regardant sa femme d'un air diplomatique, le Valenod est tout fier des deux beaux normands qu'il vient d'acheter pour sa calèche. Elle avait un certain air de simplicité, et de la jeunesse dans la démarche, aux yeux d'un parisien, cette grâce naïve, pleine d'innocence et de vivacité, serait même allée jusqu'à rappeler des idées de douce volupté. Monsieur Valenod, le riche directeur du dépôt, passait pour lui avoir fait la cour, mais sans succès, ce qui avait jeté un éclat singulier sur sa vertu, car ce monsieur Valenod, grand jeune homme, taillé en force, avec un visage coloré et de gros favoris noirs, était un de ces êtres grossiers, effrontés et bruyants, qu'en province on appelle de beaux hommes. Elle supposait sans se le dire qu'entre mari et femme il n'y avait pas de plus douces relations. Puisque Sorel n'est pas ravi et comblé de ma proposition, comme naturellement il devrait l'être, il est clair, se dit-il, qu'on lui a fait des offres d'un autre côté, et de qui peuvent-elles venir, si ce n'est du Valenod. A peine Julien fut-il à terre, que le vieux Sorel, le chassant rudement devant lui, le poussa vers la maison. Une taille svelte et bien prise annonçait plus de légèreté que de vigueur. Mais tu l'auras regardée, vilain effronté. Animal, qui te parle d'être domestique, est-ce que je voudrais que mon fils fût domestique. Cette demande déconcerta le vieux Sorel, il sentit qu'en parlant il pourrait commettre quelque imprudence, il s'emporta contre Julien, qu'il accabla d'injures, en l'accusant de gourmandise, et le quitta pour aller consulter ses autres fils. Après les avoir longtemps regardés, Julien, voyant qu'il ne pouvait rien deviner, alla se placer de l'autre côté de la scie, pour éviter d'être surpris. Mais alors plus d'avancement, plus d'ambition pour moi, plus de ce bel état de prêtre qui mène à tout. A la fin, voyant qu'il n'y avait décidément plus rien à gagner, il se retira. Il ne lui restera que ce que je viens de trouver tout fait chez le tailleur, et dont je l'ai couvert. Le joli petit prêtre, dit tout haut la cuisinière, bonne fille fort dévote. Le petit Stanislas, tout fier, lut tant bien que mal le premier mot d'un alinéa, et Julien dit toute la page. Sans qu'elle daignât le dire à personne, un accès de fièvre d'un de ses fils la mettait presque dans le même état que si l'enfant eût été mort. Je pensais, monsieur, lui dit-il un jour, qu'il y aurait une haute inconvenance à ce que le nom d'un bon gentilhomme tel qu'un Rênal parût sur le sale registre du libraire. Pour éviter tout sujet de triomphe au parti jacobin, dit le jeune précepteur, et cependant me donner les moyens de répondre à monsieur Adolphe, on pourrait faire prendre un abonnement chez le libraire par le dernier de vos gens. Le plaisant, avec tant d'orgueil, c'est que souvent il ne comprenait absolument rien à ce dont on parlait. Il y a cinquante-six ans sonnés que je suis curé de Verrières, et cependant, suivant toute apparence, je vais être destitué. J'entrevois avec peine, au fond de votre caractère, une ardeur sombre qui ne m'annonce pas la modération et la parfaite abnégation des avantage terrestres nécessaires à un prêtre, j'augure bien de votre esprit, mais, permettez-moi de vous le dire, ajouta le bon curé, les larmes aux yeux, dans l'état de prêtre, je tremblerai pour votre salut. Madame de Rênal crut sincèrement qu'elle allait devenir folle, elle le dit à son mari, et enfin tomba malade. Une chose singulière, qui trouvera peu de croyance parmi nous, c'était sans intention directe que madame de Rênal se livrait à tant de soins.

Figure 6.15 : Phrases éparses du roman « Le Rouge et le Noir » *mal* désambiguïsées par l'algorithme du Chapitre 4 (taux de succès d'uniquement 80% sur les homographes). Les homographes sont surlignés en rouge. Ceux qui ont été incorrectement désambiguïsés par l'algorithme du Chapitre 4 sont soulignés

6.3.3. Texte aléatoire automatiquement lemmatisé

Le texte aléatoire automatiquement lemmatisé présenté à la Section 5.6.2 représente le principal texte auquel les outils de lemmatisation sont soumis. Les paramètres qui ont guidé sa construction ont été fournis aux Tableaux 5.35 à 5.37. Ce texte comporte 5000 phrases et un total de 77 814 mots. Un extrait de ce texte est affiché à l'Annexe I, et ne sera donc pas affiché de nouveau ici. Pour chacun des mots de ce texte, le mot lui-même, son lemme, sa classe grammaticale (un chiffre de 1 à 9) et son étiquette morpho-syntaxique détaillée (Figure 5.33) sont fournis, pour comparaison avec les résultats des différents outils de lemmatisation. Le nombre élevé de phrases et de mots sert à confirmer que tel que désiré, de longs textes peuvent être effectivement générés dans le but d'évaluer les outils de lemmatisation sur une base statistique suffisante. Ce texte est au cœur de ce mémoire, puisque sa génération représentait l'objectif principal de ce projet de recherche.

Cependant, tel que mentionné plus tôt, seuls deux des outils décrits à la Section 6.1 seront soumis à ce texte aléatoire dans son *intégralité*, soit l'outil TreeTagger (Section 6.1.1) et l'outil développé pour ce projet (Section 6.1.4). L'information fournie en sortie par ces deux outils se prête bien à la comparaison, mot à mot, avec le détail fourni avec le texte aléatoire automatiquement lemmatisé. L'outil Cordial quant à lui, dont l'analyse ne s'effectue qu'une seule phrase à la fois et dont les résultats doivent être retranscrits manuellement, ne sera soumis qu'à un court extrait du texte aléatoire automatiquement lemmatisé (les 64 premières phrases). Cette limitation n'en est pas une reliée au générateur de textes automatiquement lemmatisés, mais plutôt à une faiblesse de l'outil commercial Cordial. Finalement, l'outil de lemmatisation simple de Jérôme Pasquelin (Section 6.1.3) ne sera pas soumis aux textes automatiquement lemmatisés, puisque cet outil ne fournit qu'une analyse globale du texte, et non une analyse mot par mot. La performance de cet outil, qui ne peut se faire qu'indirectement, par inférence, sur la base des résultats globaux, sera évaluée principalement sur la base des courts textes (Section 6.3.1).

6.3.4. Salade de mots

Tel qu'on l'a décrit à la Section 5.6.4, une « salade de mots » a été bâtie sur la base du texte aléatoire de la Section 5.6.2, comprenant exactement les mêmes mots et lemmes, mais replacés dans un ordre complètement aléatoire. Cette salade de mots est donc composée de « phrases » représentant une suite de mots sans aucun lien entre eux, sans aucune syntaxe, et évidemment complètement vides de sens. Tout comme pour le texte aléatoire de la Section 5.6.2, la salade de mots contient 5000 « phrases » et 77 814 mots.

Tel que mentionné à la Section 3.5.6, l'unique objectif de la salade de mots est de vérifier si un outil de lemmatisation utilise des indices syntaxiques ou non, lors de la lemmatisation d'un texte. Si de tels indices ne sont pas utilisés, autrement dit, si chaque mot composant le texte est analysé par lui-même sans considérer les mots qui l'entourent, on peut s'attendre à une performance de lemmatisation similaire pour le texte aléatoire de la Section 6.3.3 et pour la salade de mots. Si au contraire des indices syntaxiques sont utilisés, la performance de l'outil devrait être bien inférieure pour la salade de mots.

Les différents outils ne pourront pas être classés en ordre de performance sur la base de la salade de mots. En effet, le meilleur outil de lemmatisation pourrait fort possiblement offrir une très faible performance lorsque soumis à une salade de mots, ce qui ne témoigne en rien de la pertinence et l'efficacité de l'outil. On ne se servira de la salade de mots que pour comparer un outil de lemmatisation avec lui-même.

Cependant, tout comme pour le texte aléatoire de la section précédente, seuls deux des outils décrits à la Section 6.1 seront soumis à cette salade de mots de 5000 phrases, soit l'outil TreeTagger (Section 6.1.1) et l'outil développé dans le cadre de ce mémoire (Section 6.1.4). En

effet ces deux outils se prêtent bien à l'analyse de longs textes. Pour l'outil Cordial, puisque celui-ci ne fournit l'analyse qu'une phrase à la fois et puisque les résultats doivent être transcrits manuellement mot par mot, une bien plus courte salade de mots sera utilisée (25 phrases). Cependant, on ne pourra pas tout simplement utiliser les 25 premières phrases de la salade de mots originale de 5000 phrases, puisqu'on doit s'assurer de comparer deux textes (la salade elle-même, ainsi que le texte ordonné dont elle est issue) composés *exactement* des mêmes mots. En effet, de façon générale, les mots des 25 premières phrases des deux textes ne concordent pas, puisque la salade de mots puise des mots au hasard dans l'entièreté du texte (5000 phrases). Pour contourner ce problème, un tout nouveau texte aléatoire automatiquement lemmatisé comprenant 25 phrases a été bâti, sur la base des mêmes paramètres. Et une nouvelle salade de mots a été générée sur la base de ce texte de 25 phrases. On s'assure ainsi que les mêmes mots seront utilisés dans les deux textes pour évaluer l'outil Cordial.

Pour l'outil web de Jerôme Pasquelin, ce même court texte de 25 phrases sera utilisé, mais seules les statistiques globales de lemmatisation seront comparées, puisque cet outil ne fournit pas d'information pour chaque mot individuel en sortie.

6.4. Performance des différents outils de lemmatisation

La performance des outils de lemmatisations existants sera présentée un type de texte à la fois (courts textes, extraits du roman « Le Rouge et le Noir », texte aléatoire automatiquement lemmatisé, et salade de mots) aux Sections 6.4.1 à 6.4.4. À la Section 6.4.5, on offre un résumé de la performance de chaque outil. Il est à noter que la performance ne sera ici évaluée qu'en ce qui concerne la classe grammaticale assignée à chaque mot, parmi les neuf possibilités du Tableau 3.1 (verbe, adjectif, nom, etc.). Une évaluation plus approfondie, basée sur les sous-classes du Tableau 3.2, ne sera pas effectuée, car au-delà de la portée limitée de cette évaluation. On ne vérifiera pas non plus si les temps et personnes de verbes ont été bien assignés, de même que le genre et le nombre des noms et adjectifs.

6.4.1. Performance pour les courts textes avec buts précis

Le tout premier « court » texte, affiché précédemment à la Figure 6.10, comprend quatre phrases composées de 87 mots. Mais surtout, il ne comprend aucun homographe. L'outil de lemmatisation développé dans ce projet de recherche offre, *par défaut* une performance de lemmatisation de 100% pour tout texte ne comprenant aucun homographe, dans la mesure où tous les mots utilisés apparaissent dans les banques de mots, ce qui est le cas ici. En effet, comme une seule classe grammaticale est possible pour chacun des mots de ce texte, et comme aucune projection n'est nécessaire pour cet outil (Section 6.2.4), aucune ambiguïté n'est possible.

Pour l'outil TreeTagger, trois erreurs ont été recensées, en lien avec les mots « chaque », « ce » et « cet », qui ont été considérés par TreeTagger comme des pronoms. Tel que mentionné au Tableau 3.1, un pronom remplace généralement un groupe nominal, ce qui n'est pas le cas avec ces trois mots, qui sont tous suivis d'un nom commun (« chaque membre », « ces immenses bolides », « cet aspect »). Mais comme ces mots sont systématiquement classifiés comme des pronoms par TreeTagger, l'erreur est au niveau de l'association de mots à des classes précises, et non au niveau de l'algorithme de lemmatisation en tant que tel. Pour ce qui est de l'outil Cordial, aucune erreur de lemmatisation n'a été observée dans ce court texte sans homographes.

L'analyse de la performance de l'outil-web de Jérôme Pasquelin est plus difficile à effectuer, puisque les résultats ne sont pas fournis mot par mot. La Figure 6.9 affiche en effet un exemple de résultats fournis en sortie par cet outil, qui se limitent à la fréquence de certains lemmes. En entrant les quatre phrases de ce texte l'une après l'autre et en extrayant l'information de l'histogramme et des « remplacements » de lemmes fournis en sortie, il a été possible de déduire

les prédictions de lemmes pour 51 mots parmi les 87 composant ces quatre phrases (seulement 59% des mots). Et parmi ces 51 mots, on a recensé 5 erreurs de lemmatisation, pour une performance de lemmatisation de 90% pour ces 51 mots. Les 5 erreurs sont résumées ici :

- Le mot « du » associé au verbe « devoir » à trois reprises (le participe passé « dû » nécessite pourtant un accent circonflexe)
- L'adverbe « ne » associé au verbe « naitre » à deux reprises (le participe passé « né » nécessite pourtant un accent aigu)

Il semble donc que l'outil de Jérôme Pasquelin ne distingue pas les mots partageant la même graphie sauf pour les accents (« du » vs. « dû », et « ne » vs. « né »).

Le texte suivant est un autre court texte, mais ne comprenant cette fois *que des homographes*. Ce texte comporte 5 phrases et 89 mots. Pour l'outil TreeTagger, on parle de 22 erreurs, tandis que pour Cordial, on a recensé 11 erreurs. Pour l'outil développé pour ce projet, 20 erreurs ont été répertoriées. Pour l'outil de Jérôme Pasquelin, il a fallu une fois de plus procéder par inférence, avec l'information limitée fournie par l'histogramme et la liste de remplacement de lemmes. Parmi les 89 mots, on n'a pu identifier des lemmes que pour 54 d'entre eux (donc 61% des mots). On a recensé 15 erreurs parmi ces 54 mots. Tous ces résultats sont résumés au Tableau 6.2, accompagnés d'une description des erreurs observées pour chaque outil.

Il faut toutefois préciser que l'outil de Jérôme Pasquelin ne fournit que le lemme, et non la classe grammaticale. Par exemple, tandis qu'il associe correctement les mots « le », « la » et « l' » au lemme « le », il n'y a aucun moyen de savoir si l'outil a correctement identifié si ces homographes étaient des déterminants ou des pronoms, ou même tenté de le faire. La note de 72% octroyée au Tableau 6.2 pour cet outil est donc généreuse. On constate pour cet outil, que dès qu'un homographe peut être un verbe, il est classifié comme étant un verbe. Par exemple, les noms « domestique », « porte » et « table » ont été interprétés comme des formes verbales de « domestiquer », « porter » et « tabler ». De la même façon, l'homographe « plus » a été incorrectement classifié comme une forme du verbe « plaire », plutôt qu'un adverbe et la préposition « entre » comme une forme du verbe « entrer ».

On note qu'on retrouve pour TreeTagger, Cordial et l'outil développé pour ce projet, quelques erreurs en lien avec l'homographe « que », qui est souvent incorrectement interprété (conjonction ou pronom relatif). Ces trois outils ont aussi incorrectement associé les mots « paysan » et « contrôle » dans la première phrase comme étant un adjectif suivi d'un nom, plutôt qu'un nom suivi d'un verbe. Sans grande surprise, puisqu'il peut être associé à 5 classes grammaticales, le mot « tout » a aussi engendré quelques erreurs pour ces trois outils. Mais globalement, c'est Cordial qui a le mieux performé. En particulier, Cordial est le seul outil qui a correctement identifié le mot « La » comme étant un pronom dans le segment « La voyant partie ». C'est aussi le seul outil qui a correctement déterminé que dans le contexte de la phrase, « Rose » était un nom et non un adjectif. Pour ce qui est de l'outil développé pour le projet actuel, on constate que sa performance (78% pour ce court texte) est bien plus faible que sa performance d'environ 95% dénotée à la Section 5.3.2. Cette bien plus faible performance est due au fait qu'avec autant d'homographes dans une même phrase, les caractéristiques de chaque mot lors de l'application de l'algorithme d'apprentissage machine ne peuvent être correctement évaluées. Il s'agit évidemment ici d'un cas extrême où tous les mots d'une phrase sont des homographes, une situation qui ne se retrouve que très rarement dans le cas général.

Tableau 6.2 : Performance des outils de lemmatisation existants pour un court texte ne comprenant que des homographes avec description des erreurs observées

Outil	# erreurs et %	Exemples d'erreurs – segments de phrases. Mots mal classés **soulignés et gras.** Classification correcte en vert, classification erronée en rouge
Tree Tagger	22/89 (75%)	• (…) la porte **que** *(pronom relatif / conjonction)* le **paysan** *(nom / adjectif)* **contrôle** *(verbe / nom)* est **tout** *(adverbe / pronom)* **droit** *(adverbe / nom)* sortie de la table de **dîner** *(nom / verbe)*. • (…) pas plus de **neuf** *(déterminant / adjectif)* minutes (…) • **Rose** *(nom / adjectif)* est vite allée de nouveau **dire** *(verbe / nom)* **qu'** *(conjonction / pronom relatif)* être **dessous** *(préposition / adverbe)* le mort nuit plus que **d'** *(préposition / non classé)* être assis (…) • Dites-leur **que** *(conjonction / pronom relatif)* **leur** *(déterminant / pronom)* **mépris** *(nom / verbe)* fait sourire **tout** *(déterminant / pronom)* libéral (…) • **La** *(pronom / déterminant)* voyant **partie** *(adjectif / nom)* vers le bas, **l'** *(déterminant / non classé)* **autre** *(adjectif / nom)* **personne** *(nom / adverbe)* reste étendue de **tout** *(déterminant / pronom)* son long.
Cordial	11/89 (88%)	• (…) la porte **que** *(pronom relatif / conjonction)* le **paysan** *(nom / adjectif)* **contrôle** *(verbe / nom)* est tout **droit** *(adverbe / adjectif)* sortie de la table de **dîner** *(nom / verbe)*. • Le commandant demande que le **bien** *(adverbe / nom)* fort **courant** *(nom / adjectif)* de la chute (…) • (…) fait sourire **tout** *(déterminant / adverbe)* **libéral** *(nom / adjectif)* **que** *(pronom relatif / conjonction)* porte ton complet. • (…) l'autre personne reste étendue de **tout** *(déterminant / adjectif)* son long.
Projet actuel	20/89 (78%)	• Le vieux **domestique** *(nom / adjectif)* **fou** *(adjectif / nom)* dit que (…) • (…) la porte que le **paysan** *(nom / adjectif)* **contrôle** *(verbe / nom)* est **tout** *(adverbe / déterminant)* droit **sortie** *(adjectif / nom)* (…) • Le commandant **demande** *(verbe / nom)* **que** *(conjonction / pronom relatif)* le **bien** *(adverbe / nom)* fort **courant** *(nom / adjectif)* de la chute **demeure** *(verbe / nom)* pas **plus** *(adverbe / verbe)* (…) • **Rose** *(nom / adjectif)* est vite **allée** *(verbe / nom)* de nouveau **dire** *(verbe / nom)* **qu'** *(conjonction / pronom relatif)* être dessous le mort (…) • Dites-leur que leur **mépris** *(nom / adjectif)* fait sourire (…) • **La** *(pronom / déterminant)* **voyant** *(verbe / nom)* partie vers le bas, l'autre personne **reste** *(verbe / nom)* étendue (…)
J. Pasquelin	15/54 (72%)	• Le vieux **domestique** *(nom / verbe)* fou dit que la **porte** *(nom / verbe)* que le paysan contrôle est tout droit sortie de la **table** *(nom / verbe)* de dîner. • Le **commandant** *(nom / verbe)* demande que le bien fort **courant** *(nom / verbe)* de la **chute** *(nom / verbe)* demeure pas **plus** *(adverbe / verbe)* de neuf **minutes** *(nom / verbe)* (…) • **Rose** *(nom / verbe)* est vite allée de nouveau dire qu'être dessous le **mort** *(nom / verbe)* nuit **plus** *(adverbe / verbe)* que d'être assis **entre** *(préposition / verbe)* les **livres** *(nom / verbe)* (…) • (…) leur **mépris** *(nom / verbe)* fait sourire (…) • (…) partie **vers** *(préposition / nom)* le bas (…)

Le court texte suivant, affiché précédemment à la Figure 6.12, met l'emphase sur des homographes de même classe grammaticale, soit les mots « suis », « fils » et « convient ». L'analyse suivante ne s'attardera qu'à la lemmatisation de ces mots précis dans le texte.

L'outil TreeTagger ne permet pas de désambiguïser des homographes de même classe grammaticale. Dans tous les cas impliquant les mots « suis », « fils » et « convient », toutes les options possibles de lemmes sont fournies, sans en favoriser une en particulier. Le Tableau 6.3 illustre ce qui est fourni en sortie par TreeTagger pour ces trois mots. Il en va de même pour l'outil développé pour ce projet. Toutes les options d'homographes sont aussi fournies en sortie, comme on peut le voir au Tableau 6.4.

L'outil de Jérôme Pasquelin quant à lui fournit toujours une réponse unique. Pour l'homographe « suis », cet outil donne toujours le lemme « suivre ». Pour l'homographe « fils », cet outil donne toujours le lemme « fil ». Pour l'homographe « convient », le lemme en sortie est toujours « convier ». On constate donc que cet outil ne considère jamais le contexte dans lequel ces mots sont utilisés, fournissant tout de même une réponse en particulier, malgré l'ambiguïté non résolue.

Les résultats pour l'outil Cordial sont fournis au Tableau 6.5. On constate que cet outil applique en premier lieu les règles de grammaire. En effet, avec « tu me suis », c'est le verbe « suivre » qui est fourni en sortie, le seul possible à la deuxième personne du singulier. Et pour tous les cas où des indices du singulier précèdent le mot « fils », c'est le lemme « fils » qui est prédit (et non le lemme « fil »). Pour la phrase « ils me convient à une rencontre », il est par contre étonnant que Cordial n'ait pas correctement associé le mot « convient » au lemme « convier », puis qu'à la troisième personne du pluriel (indice « ils »), le verbe « convenir » aurait été conjugué en « conviennent ». En l'absence d'indice syntaxique (marque du singulier ou du pluriel), Cordial prédit parfois le bon lemme, parfois non. Aussi, on peut reprocher à Cordial de déterminer une classe grammaticale quand aucun indice n'est fourni, comme dans les phrases « Je suis ton père » et « J'ai perdu mes fils ». Mais toujours est-il que Cordial est le seul outil parmi les quatre testés qui utilise parfois des indices syntaxiques pour désambiguïser les homographes de même classe grammaticale.

Tableau 6.3 : Information en sortie de l'outil TreeTagger pour des homographes de même classe grammaticale. Exemples des homographes « suis », « fils » et « convient »

Homographe	Lemmes
suis	suivre | être
fils	fil | fils
convient	convenir | convier

Tableau 6.4 : Information en sortie de l'outil développé pour ce projet pour des homographes de même classe grammaticale. Exemples des homographes « suis », « fils » et « convient »

Homographe	Information en sortie
suis	être Verbe suivre Verbe
fils	fils Nom fil Nom
convient	convenir Verbe convier Verbe

Tableau 6.5 : Lemmes prédits par l'outil Cordial pour les homographes de même classe grammaticale

Phrase complète	Correct	Prédit
Je suis ton père.	?	être
Tu es mon fils, donc je suis ton père.	fils	fils
	être	être
Comme il est parti le premier, je suis ton père.	suivre	être
Tu me suis et je suis ton père.	suivre	suivre
	suivre	être
Les fils sont entremêlés.	fil	fil
Les fils de la voisine sont venus l'aider.	fils	fil
Le fils de la voisine est électricien.	fils	fils
Les fils de la voisine sont électriciens.	fils	fil
Comme fils, il ne se fait pas mieux, dit sa mère.	fils	fils
Comme fils, j'ai utilisé les plus résistants pour brancher mon appareil.	fil	fils
J'ai perdu mes fils.	?	fils
Cela me convient.	convenir	convenir
Ils me convient à une rencontre.	convier	convenir

On constate donc que des deux quatre outils (TreeTagger et l'outil du projet actuel) ne cherchent pas à désambiguïser les homographes de même classe grammaticale, fournissant en sortie toutes les possibilités. Les deux autres outils fournissent quant à eux une réponse unique. Dans le cas de l'outil de Jérôme Pasquelin, pour tout homographe donné, le lemme fourni en sortie est systématiquement le même, sans aucune justification particulière et donc sans considérer le contexte dans lequel le mot est utilisé. Par exemple, puisque le verbe « suivre » est préféré au verbe « être » pour la forme « suis », on peut en conclure que cet outil ne se base pas non plus sur la fréquence relative des mots dans le lexique, puisque le verbe être est bien plus courant. Parmi ces outils, seul Cordial donc tente de désambiguïser ces homographes, sans le faire de façon aveugle et systématique. Mais hélas, les résultats ne sont pas toujours corrects, comme on l'a vu au Tableau 6.5.

Finalement, le dernier court texte exploré s'attarde à des phrases à double sens, mais impliquant cette fois des homographes faisant intervenir différentes classes grammaticales (texte de la Figure 6.13). On retrouve deux segments de phrase (« le boucher sale la coupe » et « la belle ferme le voile ») apparaissant avec ou sans indices permettant de déterminer lequel de deux sens possibles est le plus approprié. Les résultats obtenus avec les quatre outils sont affichés au Tableau 6.6, mais se limitant aux seuls homographes correspondant à ces deux segments. Dans ce tableau, on affiche la phrase, puis la classe grammaticale correcte (« vraie »), puis les classes grammaticales prédites par les quatre outils pour chacun des homographes affichés.

On constate en premier lieu que pour les segments de phrase employés sans autre indice sémantique, aucune « vraie » classe ne peut être déduite, puisque les deux sens possibles sont tout autant probables, ce qui explique la présence de points d'interrogation dans cette colonne. Pour les phrases dans lesquelles des indices sémantiques ont été introduits, il devient possible de déterminer quelles sont les classes grammaticales correctes pour les homographes, en fonction du seul sens compatible avec les indices. La colonne « vraie » inclut donc des résultats.

Tableau 6.6 : Prédictions de classes grammaticales pour phrases à double sens (Ver=verbe, Dét = déterminant, Adj = adjectif, Nom= nom)

		Classes grammaticales				
Phrase	Mots	Vraie	Tree Tagger	Cordial	Pasquelin	Projet actuel
Le boucher sale la coupe.	sale	?	Ver	Ver	Ver	Adj
	la	?	Dét	Dét	?	Dét
	coupe	?	Nom	Nom	Ver	Nom
La pièce de viande, le boucher sale la coupe.	sale	Adj	Ver	Ver	Ver	Adj
	la	Pro	Dét	Dét	?	Dét
	coupe	Ver	Nom	Nom	Ver	Nom
Maintenant qu'il l'a préparée, le boucher sale la coupe.	sale	Ver	Ver	Adj	Ver	Adj
	la	Dét	Dét	Dét	?	Dét
	coupe	Nom	Nom	Nom	Ver	Nom
La belle ferme le voile.	belle	?	Adj	Nom	Adj	Adj
	ferme	?	Adv	Ver	Ver	Adj
	le	?	Dét	Dét	?	Dét
	voile	?	Nom	Nom	Ver	Nom
N'appréciant pas qu'il soit ouvert, la belle ferme le voile.	belle	Nom	Adj	Nom	Adj	Adj
	ferme	Ver	Adv	Ver	Ver	Adj
	le	Dét	Dét	Dét	?	Dét
	voile	Nom	Nom	Nom	Ver	Nom
Il veut admirer le paysage, mais la belle ferme le voile.	belle	Adj	Adj	Nom	Adj	Adj
	ferme	Nom	Adv	Ver	Ver	Adj
	le	Pro	Dét	Dét	?	Dét
	voile	Ver	Nom	Nom	Ver	Nom

Les prédictions offertes par les quatre outils sont ensuite affichées selon un code de couleurs. Les prédictions correctes sont en vert, les prédictions incorrectes en rouge, et finalement la couleur noire est utilisée pour les cas où il n'existe pas de classe grammaticale correcte (« vraie ») car l'ambiguïté persiste.

On constate en premier lieu que pour tous les outils, sauf Cordial, les prédictions de classe grammaticale n'ont pas été influencées par les indices sémantiques, car les résultats sont les mêmes pour les trois variations de chaque segment de phrase. Dans le cas de Cordial, il n'y a qu'un seul mot pour lequel la prédiction a varié : « sale ». Malheureusement, dans les deux cas où une classe correcte unique prévaut, les prédictions de Cordial se sont avérées incorrectes pour ce mot.

Dans le cas de l'outil de Jérôme Pasquelin, il faut se rappeler, tel que mentionné précédemment, que cet outil ne fournit en sortie que les lemmes, et non les classes grammaticales. Il n'est donc pas possible avec cet outil de déterminer la classe grammaticale des homographes « la » et « le »,

ce qui explique la présence de points d'interrogation dans la colonne « Pasquelin » pour ces deux mots.

En résumé, le Tableau 6.6 nous indique qu'aucun des quatre outils ne s'est avéré efficace pour interpréter le sens d'une phrase, dans le but de faciliter la désambiguïsation des homographes. En ce qui concerne l'outil développé pour ce projet, cette faiblesse était attendue, puisqu'aucun algorithme d'analyse sémantique n'a été mis au point. Cependant, pour les autres outils, la présence ou non de tels algorithmes restait à prouver, indirectement, par inférence.

6.4.2. Performance pour les courts extraits du roman « Le Rouge et le Noir »

Tel que décrit à la Section 6.3.2, les outils de lemmatisation ont été confrontés à deux groupes de phrases du roman « Le Rouge et le Noir ». Le premier groupe ne comporte que des phrases qui ont été parfaitement lemmatisées par l'outil développé pour le projet actuel. Le deuxième groupe de phrases quant à lui regroupe un certain nombre de phrases de ce même roman qui n'ont pas été aussi bien lemmatisées.

Il faut préciser qu'une lemmatisation manuelle a dû être effectuée pour obtenir les classes grammaticales « réelles » pour ces deux textes, mais cette lemmatisation manuelle avait déjà été effectuée dans le cadre de l'analyse du Chapitre 5. Toutefois, il est à noter que l'outil de lemmatisation de Jérôme Pasquelin, dont la performance limitée a été discutée aux sections précédentes et qui ne fournit pas en sortie une analyse mot par mot ni les classes grammaticales, n'a pas été évalué dans cette section.

Le premier texte (parfaitement lemmatisé par l'outil actuel) comporte un total de 757 mots. Le Tableau 6.7 illustre la performance de l'outil TreeTagger lors de la lemmatisation de ce texte. À la première colonne, on affiche le nombre total d'erreurs identifiées, soit 15, ainsi que la performance associée (98.0%). Ce tableau fournit ensuite le détail des 15 erreurs répertoriées, en affichant le segment de la phrase qui les contient. Le mot mal lemmatisé y est souligné en gras, suivi en vert de la classe grammaticale correcte, puis en rouge de la classe grammaticale erronée prédite par l'outil TreeTagger. Le Tableau 6.8 fournit cette même information, mais cette fois pour l'outil Cordial. On a identifié pour celui-ci 11 erreurs, pour une performance de 98.5%. Aucun tableau n'est fourni concernant l'outil de lemmatisation développé pour ce projet, puisque par défaut, sa performance pour ce texte est de 100%, puisque c'est sur cette base que ces phrases ont été sélectionnées.

Dans le cas de TreeTagger, les types d'erreurs sont assez variés. L'outil n'a par exemple pas réussi à identifier le nom propre « Verra » et l'a confondu avec une forme du verbe « voir ». À trois reprises, des noms communs ont été mal identifiés. Pour « premiers froids », TreeTagger a erronément identifié « premiers » comme étant le nom et « froids » comme étant l'adjectif. TreeTagger a aussi incorrectement associé le nom « voyageur » à un adjectif, sans doute influencé par l'erreur au mot suivant où le verbe « demande » a été associé à un nom. Finalement, le nom « pas » a été incorrectement classifié comme un adverbe, plutôt que comme un nom.

Le mot « que » et sa forme associée « qu' » ont été incorrectement classifiés en deux occasions, une fois dans le cas d'une conjonction, et l'autre dans le cas d'un pronom relatif. On se rappelle que cet homographe avait été difficile à désambiguïser par l'algorithme développé pour le projet actuel. La préposition « pour » a quant à elle été incorrectement classifiée comme une conjonction à deux reprises. Le dictionnaire en ligne Usito (2024) n'associe en effet cet homographe qu'à une préposition ou un nom commun.

Pour ce qui est de l'outil Cordial, on remarque en premier lieu qu'il a commis la même erreur de lemmatisation que TreeTagger pour le verbe « demande », incorrectement associé à un nom.

Tableau 6.7 : Performance de l'outil TreeTagger lorsque confronté à des phrases du roman « Le Rouge et le Noir » parfaitement désambiguïsées par l'outil développé pour le projet actuel

# erreurs et %	Exemples d'erreurs – segments de phrases. Mots mal classés **soulignés et gras.** Classification correcte en vert, classification erronée en rouge
15/757 (98.0%)	• (…) les cimes brisées du **Verra** *(nom / verbe)* se couvraient de neige dès les **premiers** *(adjectif / nom)* **froids** *(nom / adjectif)*. • (…) par une roue **que** *(pronom relatif / conjonction)* l'eau du torrent fait mouvoir. • (…) est **un** *(déterminant / nom)* de ceux qui étonnent le plus. • **Si** *(conjonction / adverbe)*, en entrant à Verrières, le **voyageur** *(nom / adjectif)* **demande** *(verbe / nom)* à qui appartient (…) • (…) il est chevalier de **plusieurs** *(déterminant / pronom)* ordres. • On trouve même au premier aspect **qu'** *(conjonction / pronom relatif)* elle réunit à la dignité du maire (…) • Le voyageur parisien est choqué d'un **certain** *(adjectif / pronom)* air de contentement (…) • (…) après avoir traversé la rue d'un **pas** *(nom / adverbe)* grave (…) • (…) heureusement **pour** *(préposition / conjonction)* la réputation de monsieur de Rênal. • (…) à **chaque** *(déterminant / pronom)* printemps, les eaux de pluie (…) • (…) ce mur **pour** *(préposition / conjonction)* lequel monsieur de Rênal a dû faire (…)

Tableau 6.8 : Performance de l'outil Cordial lorsque confronté à des phrases du roman « Le Rouge et le Noir » parfaitement désambiguïsées par l'outil développé pour le projet actuel

# erreurs et %	Exemples d'erreurs – segments de phrases. Mots mal classés **soulignés et gras.** Classification correcte en vert, classification erronée en rouge
11/757 (98.5%)	• (…) le voyageur **demande** *(verbe / nom)* (…) • (…) **tous** *(déterminant / adjectif)* les chapeaux se lèvent (…) • (…) il est chevalier de **plusieurs** *(déterminant / pronom)* ordres. • (…) on trouve au premier aspect **qu'** *(conjonction / pronom relatif)* elle réunit (…) • **Au-delà** *(non-classifié)*, c'est une ligne d'horizon (…) • (…) **l'** *(pronom / déterminant)* a fait maire de Verrières. • (…) la réputation de monsieur de Rênal comme **administrateur** *(nom / adjectif)* (…) • (…) les eaux de pluie sillonnent la promenade, **y** *(pronom / adverbe)* creusant des ravins (…) • (…) et de **trente** *(déterminant / nom)* ou quarante toises de long (…) • **Tous** *(déterminant / adjectif)* les ministres présents et passés (…)

On retrouve aussi deux occurrences du mot « tous », utilisé dans le contexte de ces deux erreurs comme déterminant (précédent un autre déterminant). On retrouve aussi une erreur reliée au mot « qu' », qui est une conjonction dans le contexte de la phrase en question. On constate aussi que le mot « trente » est classifié comme un nom par Cordial, alors que le mot « quarante » qui le suit deux mots plus loin a été classifié comme un adjectif numéral. Finalement, Cordial n'a associé aucune étiquette à la locution « au-delà ».

On constate donc que Cordial a offert une performance supérieure à TreeTagger pour ce texte. Le fait que le lemmatiseur développé pour ce projet a quant à lui obtenu une note parfaite pour ces phrases ne doit pas être tenu en compte, puisque ces phrases ont été spécialement sélectionnées pour que ce soit le cas. Il faut aussi préciser que les pourcentages de performance illustrés dans cette section se rapportent à tous les mots du texte, et non seulement aux homographes, comme on l'avait fait au Chapitre 5.

Les trois outils sont maintenant confrontés au deuxième regroupement de phrases, celui pour lequel l'outil de lemmatisation actuel n'avait pas aussi bien fait. De la même façon qu'on l'a fait aux Tableaux 6.7 et 6.8, on fournit aux Tableaux 6.9 à 6.11 les performances des trois outils (première colonne), ainsi que le détail d'erreurs de lemmatisation représentatives, suivant le même format. Par souci de concision, seulement 15 erreurs sont listées à chaque tableau, puisque celles-ci sont beaucoup plus nombreuses que pour le texte précédent.

On attribue un total de 78 erreurs à l'outil de lemmatisation développé pour ce projet, pour une performance globale de 92.1%. On a répertorié 63 erreurs pour TreeTagger, pour une performance de 93.7%. Et finalement, 36 erreurs ont été observées dans le cas de Cordial, pour une performance de 96.4%.

Dans le cas de l'outil développé pour ce projet (Tableau 6.9), on remarque certains cas où deux mots consécutifs sont mal lemmatisés, ce qui n'est pas surprenant, considérant le mode de fonctionnement de l'algorithme de désambiguïsation, décrit au Chapitre 4. Un seul exemple de l'homographe « de » est fourni au tableau, mais on a retrouvé plusieurs occurrences de cette erreur où ce mot devait être un déterminant plutôt qu'une préposition. On se rappellera aussi que l'homographe « de » avait nécessité son propre test spécialisé (voir Section 5.3.2.3), vu la difficulté à correctement le désambiguïser. On retrouve aussi plusieurs cas où un nom a été incorrectement classifié.

Dans le cas de TreeTagger (Tableau 6.10), on retrouve aussi des exemples d'erreurs impliquant les homographes « que » et « de ». Ces erreurs sont assez nombreuses dans le texte, ce qui témoigne une fois de plus de la difficulté associée à la désambiguïsation de ces homographes. On retrouve dans ce tableau aussi trois cas du lemme « le » incorrectement associé à un déterminant, alors qu'il est un pronom. Dans deux de ces trois cas, le mot suivant s'est retrouvé à être mal lemmatisé, sans doute en conséquence à l'erreur au lemme « le ». On constate aussi que les mots « fut-il » ont été classifiés comme étant un seul mot, un nom de surcroit. Quelques erreurs de ce type ont été recensées, impliquant une inversion du verbe et du sujet accompagnée d'un trait d'union, avec l'outil TreeTagger.

L'outil Cordial a mieux performé, avec environ moitié moins d'erreurs que les deux autres outils. On y retrouve tout de même des erreurs semblables, entre autres impliquant les homographes « de », « que » et des variantes de « tout ». On retrouve aussi la même erreur au verbe « sourire » observée avec l'outil développé pour ce projet, et la même erreur pour le nom « curé » que celle observée avec TreeTagger.

Le Tableau 6.12 résume quantitativement la performance des trois outils lorsque confrontés aux deux recueils de phrases du roman « Le Rouge et le Noir » discutés ici. Pour les deux textes, on compile le nombre d'erreurs répertoriées pour chaque outil, et la performance associée. Finalement, la Figure 6.16 fournit graphiquement la même information.

Tableau 6.9 : Performance de l'outil développé pour le projet actuel avec emphase sur des phrases du roman « Le Rouge et le Noir » *mal* désambiguïsées. Liste de 15 exemples d'erreurs

# erreurs et %	Exemples d'erreurs – segments de phrases. Mots mal classés **soulignés et gras.** Classification correcte en vert, classification erronée en rouge
78/993 (92.1%)	• (…) chacun de ces marteaux **fabrique** *(verbe / nom)* (…) • (…) acheté au poids de l'**or** *(nom / conjonction)* (…) • (…) il a dû lui compter **de** *(déterminant / préposition)* beaux louis d'or (…) • (…) monsieur de Rênal, **revenant** *(verbe / nom)* de l'église en costume de maire, vit de loin le **vieux** *(adjectif / nom)* Sorel entouré de ses trois fils **sourire** *(verbe / nom)* en **le** *(pronom / déterminant)* **regardant** *(verbe / adjectif)* (…) • (…) on sait que vous avez livres de **rente** *(nom / verbe)*, du bon **bien** *(nom / adverbe)* au soleil (…) • (…) un champ qui rapporte **livres** *(nom / verbe)* (…) • (…) le Valenod est **tout** *(adverbe / pronom)* **fier** *(adjectif / verbe)* (…) • (…) serait **même** *(adverbe / adjectif)* **allée** *(verbe / nom)* jusqu'à rappeler (…)

Tableau 6.10 : Performance de l'outil TreeTagger lorsque confronté à des phrases du roman « Le Rouge et le Noir » *mal* désambiguïsées par l'outil développé pour le projet actuel. Liste de 15 exemples d'erreurs

# erreurs et %	Exemples d'erreurs – segments de phrases. Mots mal classés **soulignés et gras.** Classification correcte en vert, classification erronée en rouge
63/993 (93.7%)	• (…) **certains** *(déterminant / pronom)* petits morceaux de terrain qu'ils occupent (…) • (…) il a dû lui compter **de** *(déterminant / préposition)* beaux louis d'or (…) • (…) comme on l'appelle depuis **qu'** *(conjonction / pronom relatif)* il est riche (…) • (…) le Valenod est tout **fier** *(adjectif / verbe)* (…) • (…) elle supposait sans se **le** *(pronom / déterminant)* **dire** *(verbe / nom)* (…) • (…) comme naturellement il devrait **l'** *(pronom / déterminant)* **être** *(verbe / nom)* (…) • (…) à peine Julien **fut-il** *(verbe suivi de pronom / nom)* à terre, que le vieux Sorel le chassant rudement devant lui **le** *(pronom / déterminant)* **poussa** *(verbe / nom)* vers la maison. • (…) il l'accabla d'injures en **l'** *(pronom / déterminant)* accusant (…) • (…) pour éviter tout sujet de triomphe au **parti** *(nom / adjectif)* **jacobin** *(adjectif / nom)* (…) • (…) je suis **curé** *(nom / verbe)* de Verrières (…)

Tableau 6.11 : Performance de l'outil Cordial lorsque confronté à des phrases du roman « Le Rouge et le Noir » *mal* désambiguïsées par l'outil développé pour le projet actuel. Liste de 15 exemples d'erreurs

# erreurs et %	Exemples d'erreurs – segments de phrases. Mots mal classés **soulignés et gras.** Classification correcte en vert, classification erronée en rouge
36/993 (96.4%)	• (…) les façades de presque **toutes** *(déterminant / adjectif)* les maisons (…) • (…) il a dû lui compter **de** *(déterminant / préposition)* beaux louis d'or (…) • (…) un visage coloré et **de** *(déterminant / préposition)* gros favoris noirs (…) • (…) vit de loin le vieux Sorel entouré de ses trois fils **sourire** *(verbe / nom)* en le regardant. • Elle supposait sans se **le** *(pronom / déterminant)* **dire** *(verbe / nom)* **qu'** *(conjonction / pronom)* entre mari et femme (…) • Est-ce que je voudrais que mon fils fût **domestique** *(nom / adjectif)* ? • (…) Julien dit toute la page sans **qu'** *(conjonction / pronom relatif)* elle daignât (…) • (…) le même état **que** *(pronom relatif / adverbe)* si l'enfant eût été (…) • (…) **tout** *(déterminant / adverbe)* sujet de triomphe au parti jacobin (…) • **Le** *(pronom / déterminant)* **plaisant** *(verbe / nom)*, avec tant d'orgueil, c'est que souvent il ne comprenait absolument rien (…) • (…) je suis **curé** *(nom / verbe)* de Verrières (…) • (…) et la **parfaite** *(adjectif / nom)* abnégation (…)

Considérant qu'on ne peut tenir compte de la performance parfaite (par défaut) de l'outil développé pour ce projet dans le cas du « texte bien lemmatisé », on constate que c'est l'outil Cordial qui a offert globalement la meilleure performance de lemmatisation. L'outil TreeTagger a offert une performance légèrement supérieure à l'outil développé pour ce projet, mais il faut tenir compte du fait que les phrases du deuxième regroupement avaient été sélectionnées spécifiquement selon la mauvaise performance de l'outil actuel. La comparaison directe n'est donc pas tout à fait impartiale.

Tableau 6.12 : Performance de lemmatisation de trois outils de lemmatisation confrontés à des portions du roman « Le Rouge et le Noir » bien et mal lemmatisées par l'outil du projet actuel

	Portion du texte bien lemmatisée			
# mots		Tree Tagger	Cordial	Projet actuel
757	# erreur	15	11	0
	% erreur	1.98%	1.45%	0.00%
	Performance	98.0%	98.5%	100%

	Portion du texte mal lemmatisée			
# mots		Tree Tagger	Cordial	Projet actuel
993	# erreur	63	36	78
	% erreur	6.34%	3.63%	7.85%
	Performance	93.7%	96.4%	92.1%

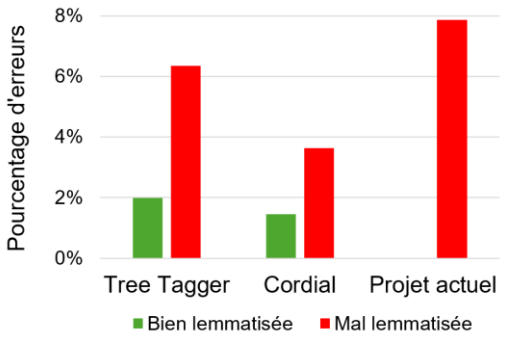


Figure 6.16 : Pourcentage d'erreur pour trois outils de lemmatisation confrontés à des portions bien et mal lemmatisées du roman « Le Rouge et le Noir » par l'outil du projet actuel

On constate surtout, comme on l'avait prévu, que TreeTagger et Cordial ont offert une meilleure performance lorsque confrontés au premier texte, que confrontés au second. Cela confirme l'hypothèse avancée à la Section 6.3.2 comme quoi si un texte est difficile à lemmatiser pour un outil en particulier, il risque de l'être aussi pour un autre outil, même si ceux-ci fonctionnent sur la base d'algorithmes différents. On avait fait une constatation semblable à la Section 6.4.1, en comparant la performance des quatre outils confrontés à de courts textes ne comprenant soit *aucun homographe*, ou ne comprenant *que des homographes*.

6.4.3. Performance pour les textes aléatoires automatiquement lemmatisés

Il est maintenant temps de confronter les outils de lemmatisation au texte qui est au cœur de ce projet, le texte aléatoire automatiquement lemmatisé, présenté une première fois à la Section 5.6.2, puis décrit à la Section 6.3.3 dans le contexte d'évaluation de lemmatiseurs existants. Ce texte comporte 5000 phrases, comprenant au total 77 814 mots. Une fois de plus, l'outil de lemmatisation de Jérôme Pasquelin n'est pas évalué, car non approprié pour l'exercice présent qui consiste à évaluer la performance des outils mot par mot, ce que ne permet pas cet outil web simple. En effet, celui-ci ne fournit des résultats que sous forme de résumés (histogrammes et tableaux de remplacement de lemmes). Comme l'outil Cordial requiert un transfert manuel des résultats de la lemmatisation, tel que mentionné à la Section 6.1.1, celui-ci n'a été soumis qu'à un court extrait du texte aléatoire décrit plus haut. En effet, seules les 64 premières phrases, comprenant 1055 mots, ont été soumises à l'analyse de Cordial. L'outil TreeTagger et l'outil de lemmatisation développé pour ce projet ont quant à eux été testés avec le texte aléatoire dans son intégralité (5000 phrases).

Dans un premier temps, on montre aux Tableaux 6.13 à 6.15, des résultats détaillés pour les trois outils testés, en se limitant pour l'instant aux 64 premières phrases du texte aléatoire automatiquement lemmatisé. Utilisant le même format que précédemment, on indique le nombre total d'erreurs et la performance de lemmatisation à la première colonne. On affiche ensuite un échantillon de 15 erreurs représentatives pour chacun des outils, utilisant le même format qu'aux tableaux précédents (segments de phrases, mot incorrectement lemmatisé souligné, code de couleurs). Cette analyse permet de vérifier si on retrouve le même type d'erreur avec ce texte automatisé qu'avec un texte littéraire authentique (le roman « Le Rouge et le Noir ») comme on l'a fait à la section précédente.

Pour ce qui est de la performance globale sur ces 64 phrases, c'est une fois de plus l'outil Cordial qui s'en tire le mieux, avec 34 erreurs, pour une performance de 96.8%. Il est suivi de l'outil de lemmatisation développé pour ce projet, avec 37 erreurs et une performance de 96.5%. L'outil TreeTagger ferme la marche, avec davantage d'erreurs, soit 56, pour une performance globale de 94.7%. Dans tous les cas, on recense un bon nombre d'erreurs en lien avec les homographes « de » et « que », tout comme on l'avait observé avec les extraits du roman « Le Rouge et le Noir ». On retrouve aussi pour les trois outils plusieurs erreurs impliquant des homographes de type « nom-adjectif », dont quelques cas d'inversion, où un nom suivi d'un adjectif sont incorrectement interprétés comme un adjectif antéposé suivi d'un nom.

Plus particulièrement pour TreeTagger (Tableau 6.13), on note plusieurs erreurs impliquant des adverbes comme « si » (incorrectement interprété ici comme une conjonction), « fort » (incorrectement interprété comme un adjectif) et « trop » (étonnamment incorrectement interprété comme un nom commun). Pour Cordial (Tableau 6.14), on affiche ici deux exemples de verbes (ou participe passé) incorrectement associés à des noms (« tombe » et « mépris »). Pour l'outil développé pour ce projet (Tableau 6.15), on observe quelques erreurs impliquant des homographes de type « déterminant-pronom », comme pour « le », « les », « laquelle ». On note aussi deux cas de l'adverbe « plus » incorrectement identifié comme une forme du verbe « plaire ».

Tableau 6.13 : Performance de l'outil TreeTagger lorsque confronté aux 64 premières phrases (1055 mots) du texte aléatoire automatiquement lemmatisé. Liste de 15 exemples d'erreurs

# erreurs et %	Exemples d'erreurs – segments de phrases. Mots mal classés **soulignés et gras.** Classification correcte en vert, classification erronée en rouge
56/1055 (94.7%)	• (…) **chaque** *(déterminant / pronom)* fagot qui ne déménagea pas (…) • Je ne me jetai pas **de** *(déterminant / préposition)* cris tellement fermés (…) • (…) ta force **qu'** *(pronom relatif / conjonction)* un quelconque plus bon confesseur (…) • Vous n'aviez pas pu reprocher d'emplois **si** *(adverbe / conjonction)* **bas** *(adjectif / adverbe)* (…) • (…) un teint presque vêtu de ce différend **équivoque** *(adjectif / verbe)* (…) • Il fallait **que** *(conjonction / pronom relatif)* vous vous eussiez consultées (…) • (…) un accueil d'une telle **surveillante** *(nom / adjectif)* plus tremblante (…) • (…) qui voyait un **premier** *(adjectif / déterminant)* pape (…) • (…) et à un tel **paysan** *(nom / adjectif)* écrasé (…) • (…) qui ira profondément faillir de **parler** *(verbe / nom)* (…) • (…) que l'aspect presque **repoussant** *(adjectif / verbe)* (…) • (…) les **trop** *(adverbe / nom)* **premiers** *(adjectif / déterminant)* papes n'avaient pas voulu (…) • (…) l'hardiesse **fort** *(adverbe / adjectif)* violente (…)

Tableau 6.14 : Performance de l'outil Cordial lorsque confronté aux 64 premières phrases (1055 mots) du texte aléatoire automatiquement lemmatisé. Liste de 15 exemples d'erreurs

# erreurs et %	Exemples d'erreurs – segments de phrases. Mots mal classés **soulignés et gras.** Classification correcte en vert, classification erronée en rouge
34/1055 (96.8%)	• Je ne me jetai pas **de** *(déterminant / préposition)* cris tellement fermés (…) • (…) parlèrent de mes **curés** *(nom / adjectif)* **vénérables** *(adjectif / nom)* (…) • (…) une objection plus soudaine **que** *(pronom relatif / adverbe)* des marquis trop pairs et un égard innocent devaient parcourir, **tombe** *(verbe / nom)* (…) • (…) vous n'aviez pas pu reprocher **d'** *(déterminant / préposition)* emplois (…) • (…) ce **témoin** *(nom / adjectif)* **noble** *(adjectif / nom)* avait dû se vouloir (…) • (…) un **jardinier** *(nom / adjectif)* **officiel** *(adjectif / nom)* de ces glaces plus entières (…) • (…) n'avoue pas le souverain plus **mépris** *(verbe / nom)* (…) • **Nulle** *(déterminant / adjectif)* passion ne voulait pas entrer ainsi (…) • (…) à laquelle des **destinées** *(nom / adjectif)* **gauches** *(adjectif / nom)* et des temps (…) • (…) un jeune hardiesse **qu'** *(pronom relatif / conjonction)* une fortune et des images plus contradictoires oublièrent (…)

Tableau 6.15 : Performance de l'outil de lemmatisation développé pour ce projet lorsque confronté aux 64 premières phrases (1055 mots) du texte aléatoire automatiquement lemmatisé. Liste de 15 exemples d'erreurs

# erreurs et %	Exemples d'erreurs – segments de phrases. Mots mal classés **soulignés et gras.** Classification correcte en vert, classification erronée en rouge
37/1055 (96.5%)	• Je ne me jetai pas **de** *(déterminant / préposition)* cris tellement fermés (…) • (…) une objection plus soudaine **que** *(pronom relatif / conjonction)* des marquis trop pairs (…) • (…) et des juges plus mortelles **que** *(pronom relatif / conjonction)* cette madame seule ne voulait pas rapidement susciter. • (…) de la tentative à **laquelle** *(pronom / déterminant)* des destinées gauches (…) • (…) nos voix qui ne câlinèrent pas **les** *(déterminant / pronom)* si fameuses madames (…) • (…) ce talent **plus** *(adverbe / verbe)* **saint** *(adjectif / nom)* d'une madame (…) • (…) **les** *(déterminant / pronom)* trop premiers papes n'avaient pas voulu voir **d'** *(déterminant / préposition)* adjointes plus occupées (…) • (…) vos **administrés** *(nom / adjectif)* **principaux** *(adjectif / nom)* qui arrivèrent (…) • (…) un chiffon qui avait accablé **de** *(déterminant / préposition)* telles prouesses (…) • (…) des surveillants très antiques **que** *(pronom relatif / conjonction)* des suretés anéanties et des témoins avaient précédés. • (…) une certaine concurrente **plus** *(adverbe / verbe)* **riche** *(adjectif / nom)* (…)

On a ensuite évalué la performance de TreeTagger et de l'outil développé pour ce projet, lorsque confrontés au texte automatiquement lemmatisé dans son entier (5000 phrases). Pour TreeTagger, on a observé un total de 3561 erreurs, pour une performance globale de 95.4%. Pour l'outil du projet actuel, on a recensé 2547 erreurs, pour une performance de 96.7%. On note en premier lieu que ces performances globales ne sont pas très différentes de ce qu'on avait observé pour l'échantillon réduit d'uniquement 64 phrases, tel qu'affiché aux Tableaux 6.13 et 6.15. Les pourcentages d'erreurs sont en effet assez semblables, comme on peut le voir au Tableau 6.16 qui résume ces résultats, et à la Figure 6.17 qui les exprime graphiquement. Ces similarités nous permettent d'émettre l'hypothèse que la performance observée pour Cordial sur les 64 premières phrases puisse être extrapolée pour le texte dans son intégralité. On peut donc, avec une certaine confiance, décréter que c'est l'outil Cordial qui a été le plus efficace des trois, lorsque confronté au texte aléatoire automatiquement lemmatisé.

On peut aussi comparer les pourcentages d'erreurs du Tableau 6.16 à ceux obtenus dans le cas des extraits du roman « Le Rouge et le Noir » au Tableau 6.12. On constate que les pourcentages d'erreurs pour le texte automatiquement lemmatisé se retrouvent à mi-chemin entre ceux obtenus pour la portion bien lemmatisée du roman, et la portion moins bien lemmatisée. On peut donc en déduire que le niveau de difficulté imposé aux lemmatiseurs est somme toute assez semblable entre les extraits du roman, et le texte aléatoire automatiquement lemmatisé. Cette observation suggère donc que le texte aléatoire automatiquement lemmatisé représente un « étalon doré » pertinent, imposant des défis de lemmatisation appropriés aux différents outils. On constate d'ailleurs qu'on retrouve des types d'erreurs semblables pour les deux types de textes (texte de littérature réel, et texte aléatoire). Il semble donc que l'objectif principal du projet ait été atteint.

Tableau 6.16 : Performance de lemmatisation de trois outils de lemmatisation confrontés à des textes aléatoires automatiquement lemmatisés

Texte de 64 phrases				
# mots		Tree Tagger	Cordial	Projet actuel
1055	# erreur	56	34	37
	% erreur	5.31%	3.22%	3.51%
	Performance	94.7%	96.8%	96.5%

Texte de 5000 phrases				
# mots		Tree Tagger	Cordial	Projet actuel
77814	# erreur	3561	non évalué	2547
	% erreur	4.58%		3.27%
	Performance	95.4%		96.7%

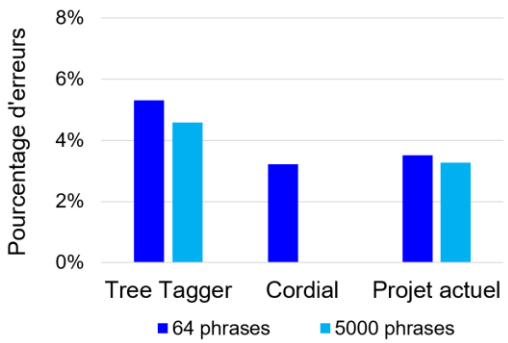


Figure 6.17 : Pourcentage d'erreur pour trois outils de lemmatisation confrontés à des textes aléatoires automatiquement lemmatisés

Plutôt que de se limiter à la performance globale de chaque outil, il est pertinent de pousser l'analyse plus loin, et d'explorer leur performance en fonction des classes grammaticales en cause, à l'aide de matrices de confusion, comme celles utilisées au Chapitre 5. Cependant, en contraste avec l'analyse du Chapitre 5, les matrices de confusion considèrent ici *tous les mots du texte*, et non seulement les homographes, car certains non homographes ont été mal classifiés par TreeTagger et Cordial.

Les Figures 6.18 à 6.20 illustrent les matrices de confusion pour les trois outils testés dans cette section. Pour TreeTagger et l'outil du projet actuel, ces matrices sont bâties sur la base de l'analyse des 5000 phrases, tandis que pour Cordial, la matrice est bâtie sur la base des 64 premières phrases seulement. Les rangées de ces matrices correspondent aux classes réelles des différents mots, alors que les colonnes correspondent aux prédictions offertes par les trois outils. Les cases en vert sur la diagonale correspondent aux prédictions correctes, alors que les cases en rouge correspondent aux prédictions incorrectes. La toute dernière colonne fournit le rappel pour chaque classe, soit la proportion de mots de chaque classe bien prédits. La toute dernière ligne fournit quant à elle la précision, soit la performance des prédictions pour chaque classe. Finalement, la performance globale, déjà rapportée aux tableaux précédents, apparait dans le coin inférieur droit pour ces trois matrices.

On remarque d'abord que la matrice de confusion pour l'outil TreeTagger comporte davantage de cases rouges, en comparaison avec les deux autres outils, ce qui implique qu'on observe pour cet outil un plus grand éventail de types d'erreurs. Mais surtout, on constate que le nombre de cases rouges est bien réduit pour Cordial, ce qui confirme une fois de plus la supériorité de cet outil, car il résulte en moins de types d'erreurs. Des efforts pour en améliorer encore davantage la performance pourraient donc être plus concentrés que pour améliorer la performance des deux autres outils. Il faut toutefois se rappeler que Cordial a été évalué ici sur la base d'un plus faible échantillon (64 phrases).

Pour Cordial, les erreurs les plus communes impliquent des déterminants incorrectement interprétés comme des prépositions, ce qui se rapporte plus spécifiquement à l'homographe « de », dont la désambiguïsation représente un défi important pour les trois outils. Il est en effet difficile de bien identifier les cas où le mot « de » est un déterminant, comme dans les segments de phrases « il a *de* gros favoris noirs » et « il ne porte pas *de* chapeau ». Sinon, Cordial confond parfois un nom avec un adjectif, ou l'inverse, incluant des cas où deux de ces types de mots se suivent.

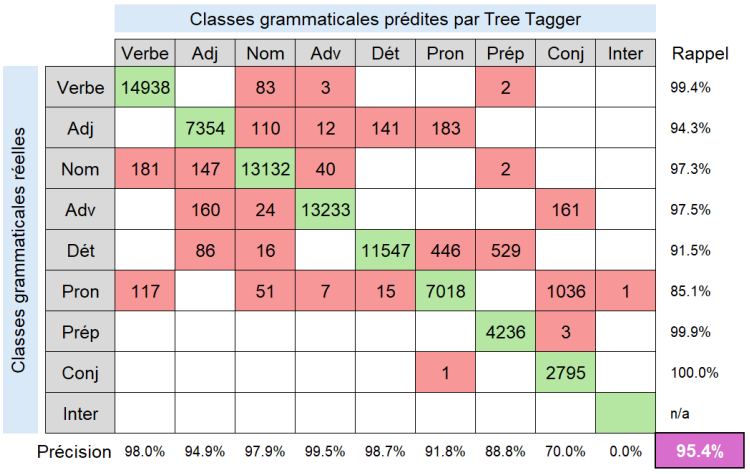


Figure 6.18 : Matrice de confusion pour l'outil TreeTagger confronté à un texte aléatoire automatiquement lemmatisé de 5000 phrases

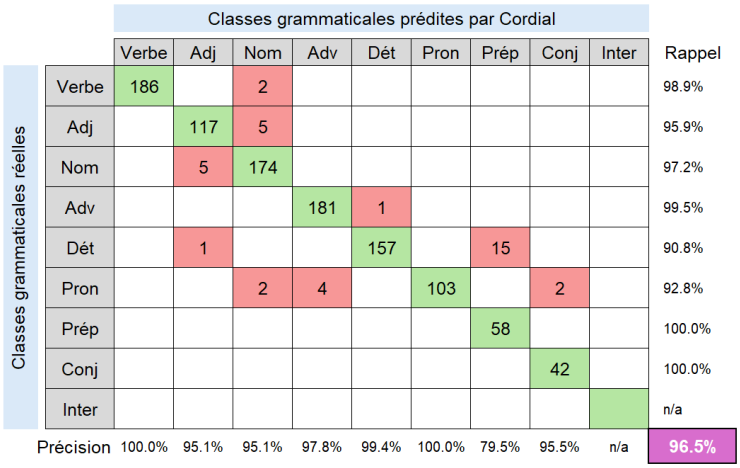


Figure 6.19 : Matrice de confusion pour l'outil Cordial confronté à un texte aléatoire automatiquement lemmatisé de 64 phrases

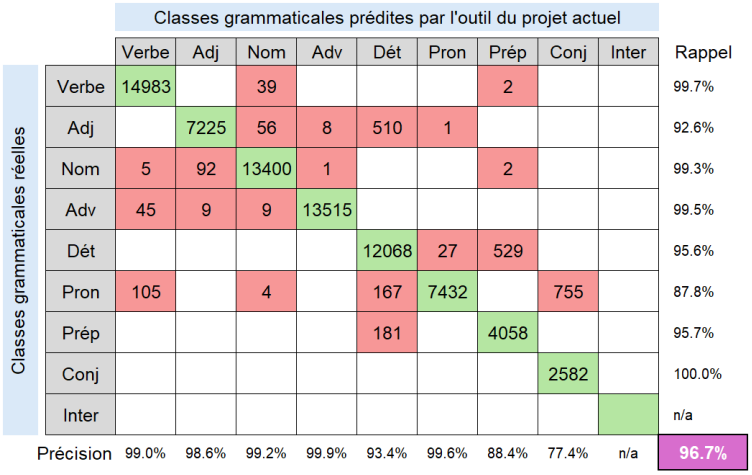


Figure 6.20 : Matrice de confusion pour l'outil de lemmatisation développé pour le projet actuel confronté à un texte aléatoire automatiquement lemmatisé de 5000 phrases

Dans le cas de TreeTagger, l'erreur la plus courante implique des pronoms incorrectement interprétés comme des conjonctions. Ceci implique l'homographe « que », reconnu comme étant difficile à désambiguïser au cours de ce projet, cet homographe ayant eu droit à son propre « test spécialisé » (Section 4.6.5) pour la désambiguïsation. Il est en effet nécessaire dans plusieurs cas d'analyser le sens de la phrase afin de bien déterminer la classe grammaticale de ce mot dans le contexte de son utilisation. On constate d'ailleurs à la Figure 6.18 une faible valeur de rappel pour les pronoms (85.1%), mais surtout, une encore plus faible valeur de précision pour les conjonctions (70.0%). En revanche, TreeTagger s'est avéré très efficace pour correctement classifier les conjonctions quand il en croise une, avec une seule erreur recensée, et donc un rappel de près de 100% pour cette classe. On en conclut donc que TreeTagger a tendance à voir plus de conjonctions qu'on en retrouve réellement dans un texte.

Sinon, TreeTagger a aussi parfois du mal à bien classifier certains déterminants. En effet, on remarque à la Figure 6.18 une valeur de rappel de 91.5% pour les déterminants, la plus faible après celle pour les pronoms. En particulier, les déterminants sont souvent incorrectement classifiés comme pronoms ou comme prépositions. Les cas impliquant les pronoms comprennent les homographes « le », « la » et « les », alors que les cas impliquant des prépositions impliquent plutôt principalement l'homographe « de », qui a causé problèmes aussi pour Cordial.

Pour ce qui est de l'outil développé pour ce projet, on dénote aussi un assez grand nombre de cases rouges, indiquant que plusieurs types d'erreurs ont été recensés, et qu'une amélioration de cet outil impliquerait donc plusieurs stratégies différentes. Tout comme pour TreeTagger, le type d'erreur le plus courant concerne les pronoms incorrectement interprétés comme des conjonctions. Ce type d'erreur contribue à la faible valeur de rappel pour les pronoms (87.8%), et la valeur encore plus faible pour la précision pour les conjonctions (77.4%). On note aussi que tout comme pour TreeTagger, le rappel pour les conjonctions est de 100%. L'outil du projet actuel a donc lui aussi tendance à surestimer la proportion de conjonctions dans un texte, au détriment des cas de l'homographe « que » pouvant être un pronom. On note ensuite un nombre d'erreurs élevé pour les déterminants incorrectement interprétés comme des prépositions (le cas « de »), avec 529 erreurs. On note aussi 181 erreurs pour le cas inverse, où la préposition « de » est incorrectement interprétée comme un déterminant. Ces résultats mettent en évidence la difficulté de lemmatiser l'homographe « de ». On retrouve finalement aussi un bon nombre d'adjectifs incorrectement interprétés comme des déterminants. La précision pour les déterminants est en effet assez faible, avec une valeur de 93.4%. Il s'agit ici d'homographes tels que « certain », « tel », et leurs variantes, dont la classe dépend fortement du contexte de leur utilisation.

Les types d'erreurs observés pour les trois outils, lorsque confrontés au texte aléatoire automatiquement lemmatisé, sont résumés au Tableau 6.17. Dans ce tableau, on ne s'attarde en effet qu'aux erreurs, donc aux cases rouges des matrices de confusion. On fournit la proportion d'erreurs associées à chaque type. Un tel tableau s'avère pratique pour prioriser les améliorations à apporter à chaque outil pour en augmenter la performance globale, puisqu'il met en évidence les plus grandes faiblesses de chacun. En effet, pour TreeTagger, on doit prioriser les erreurs de type « pronom-conjonction », « déterminant-préposition » et « déterminant-pronom ». Ces trois types d'erreurs représentent à eux seuls plus de la moitié des erreurs observées avec TreeTagger. Pour Cordial, se concentrer sur les erreurs de type « déterminant-préposition » est de loin la priorité (41% des erreurs). Pour l'outil du projet actuel, on retrouve aux deux premières places les mêmes types d'erreurs que pour TreeTagger, soit « pronom-conjonction » et « déterminant-préposition ». En troisième place, on retrouve les erreurs de type « adjectif-déterminant ». Ces trois types d'erreurs représentent 70% du total des erreurs.

Ce tableau met donc en relief la pertinence des textes aléatoires automatiquement lemmatisés pour identifier les faiblesses des outils de lemmatisation, et donc de suggérer des pistes d'amélioration. Une fois de plus, ceci confirme la pertinence du projet de recherche actuel.

Tableau 6.17 : Compilation des erreurs de prédictions pour trois outils de lemmatisation, par type de classe grammaticale réelle et prédite

Tree Tagger		
Classe Grammaticale		% des erreurs
Réelle	Prédite	100%
Pronom	Conjonction	29%
Déterminant	Préposition	15%
Déterminant	Pronom	13%
Adjectif	Pronom	5.1%
Nom	Verbe	5.1%
Adverbe	Conjonction	4.5%
Adverbe	Adjectif	4.5%
Nom	Adjectif	4.1%
Adjectif	Déterminant	4.0%
Pronom	Verbe	3.3%
Adjectif	Nom	3.1%
Déterminant	Adjectif	2.4%
Verbe	Nom	2.3%
Pronom	Nom	1.4%
Nom	Adverbe	1.1%
Adverbe	Nom	0.67%
Déterminant	Nom	0.45%
Pronom	Déterminant	0.42%
Adjectif	Adverbe	0.34%
Pronom	Adverbe	0.20%
Verbe	Adverbe	0.084%
Préposition	Conjonction	0.08%
Verbe	Préposition	0.056%
Nom	Préposition	0.056%
Pronom	Interjection	0.028%
Conjonction	Pronom	0.028%

Cordial		
Classe Grammaticale		% des erreurs
Réelle	Prédite	100.00%
Déterminant	Préposition	41%
Adjectif	Nom	14%
Nom	Adjectif	14%
Pronom	Adverbe	11%
Verbe	Nom	5.4%
Pronom	Nom	5.4%
Pronom	Conjonction	5.4%
Adverbe	Déterminant	2.7%
Déterminant	Adjectif	2.7%

Projet actuel		
Classe Grammaticale		% des erreurs
Réelle	Prédite	100.00%
Pronom	Conjonction	30%
Déterminant	Préposition	21%
Adjectif	Déterminant	20%
Préposition	Déterminant	7.1%
Pronom	Déterminant	6.6%
Pronom	Verbe	4.1%
Nom	Adjectif	3.6%
Adjectif	Nom	2.2%
Adverbe	Verbe	1.8%
Verbe	Nom	1.5%
Déterminant	Pronom	1.1%
Adverbe	Adjectif	0.35%
Adverbe	Nom	0.35%
Adjectif	Adverbe	0.31%
Nom	Verbe	0.20%
Pronom	Nom	0.16%
Verbe	Préposition	0.079%
Nom	Préposition	0.079%
Adjectif	Pronom	0.039%
Nom	Adverbe	0.039%

6.4.4. Performance pour la salade de mots

Le dernier type de texte auquel on confronte les outils de lemmatisation est la salade de mots. On se rappelle que l'unique utilité de la salade de mots est de vérifier si un outil de lemmatisation fait appel aux indices syntaxiques ou non, lors de la désambiguïsation des homographes. En effet, si pour un outil donné, une performance identique est obtenue avec un texte « ordonné » et sa salade de mots correspondante, on peut en déduire que cet outil n'utilise aucun indice syntaxique. À l'inverse, si la performance d'un outil est bien supérieure (ou à tout le moins très différente) pour le texte ordonné par rapport à la salade, on peut en déduire que l'outil en question fait appel aux indices syntaxiques, puisque la séquence des mots se retrouve à affecter les résultats.

La salade de mots ne sert pas du tout à comparer entre elles les performances des différents outils de lemmatisation. On peut d'ailleurs s'attendre à une performance assez faible, même pour les meilleurs outils de lemmatisation. La seule comparaison pertinente est d'un outil donné envers lui-même, lorsque confronté à ces deux types de textes.

On se rappelle qu'une salade de mots avait été générée à la Section 5.6.4, et que cette salade a été bâtie sur la base du texte aléatoire automatiquement lemmatisé généré à la Section 5.6.2. Ces deux textes comportent le même nombre de phrases (même si le concept de phrase est peu approprié à une salade de mots) et le même nombre de mots, par défaut, puisque la salade ne fait que « brasser » les mots du texte original pour les replacer dans un ordre tout à fait arbitraire et aléatoire.

L'outil TreeTagger et l'outil de lemmatisation développé pour ce projet ont tous deux été évalués sur la base du texte aléatoire et de sa salade correspondante décrits au Chapitre 5 ainsi qu'aux

Sections 6.3.3. et 6.3.4, soit des textes de 5000 phrases comprenant un total de 77 814 mots. En revanche, de bien plus courts textes ont été utilisés pour évaluer Cordial et l'outil de Jérôme Pasquelin. Dans le cas de Cordial, ce choix est dû au fait que les résultats de cet outil commercial doivent être manuellement transcrits pour être analysés, une tâche longue et pénible. L'outil de Jérôme Pasquelin implique quant à lui un défi différent, car on ne peut déduire les classes grammaticales des mots que par inférence, en analysant les résultats fournis de façon globale (résumé) dans des histogrammes et tableaux de remplacement. Il s'agit là encore d'une opération manuelle, qui de surcroit ne fournit que des résultats partiels, puisque disponibles que pour un sous-ensemble des mots inclus dans le texte. On a donc soumis ces deux outils à des textes (ordonné et salade de mots) d'uniquement 25 phrases.

Cependant, on ne peut se contenter d'utiliser tout simplement les 25 premières phrases du texte aléatoire automatiquement lemmatisé et de sa salade correspondante. Ceci est dû au fait que de façon générale, les mots inclus dans les 25 premières phrases de la salade de mots ne correspondent pas aux mots inclus dans les 25 premières phrases du texte original. En effet, les premières phrases de la salade incluent des mots issus de n'importe où dans le texte original, bien au-delà des 25 premières phrases. Et comme il est essentiel de comparer deux textes (ordonné et salade) comportant *exactement* les mêmes mots, il a donc fallu générer un tout nouveau texte aléatoire de 25 phrases, ainsi qu'une nouvelle salade de mots associée à ces 25 phrases. Le nouveau texte aléatoire, généré en utilisant les mêmes paramètres que pour le texte aléatoire de 5000 phrases, est illustré à la Figure 6.21. La salade correspondante est affichée à la Figure 6.22. Ces deux textes utilisent exactement les mêmes mots.

```
Des illusions d'une telle femme éblouissante, qui partaient les nouvelles
chambres et les fils absolument monotones, savaient aimer doucement les
petites discussions de cette vanité. Les nôtres n'avaient pas cru s'écrier
une telle colère bouillante et de telles politiques de cette contrainte et
de ces hauteurs convenables. Ces avis odieux et ces salaires plus pensifs te
verraient ainsi. Il fallait que nous nous fussions dites une jolie fièvre et
une femme plus fixe d'une telle discussion. Une telle persienne plus fermée
ne dut pas autrement aimer ce génie fort rempli de quelques propriétés tant
rangées et de quelques notions exagérées, auquel un suisse faillait
précisément. Tu auras baissé ce bouledogue. Vois le bénitier et les
manifestations! Le don presque anéanti et les rôles n'auraient pas ainsi
voulu jouer ce fait des camarades rassurées et des étendues pâles, qui baptisa
nos façades pénibles. Nous ne nous crûmes pas timidement en celles-là. Il
fallait que nous nous soyons sentis cette pompe possible et ces vêtements.
Notre gardienne plus marquée, que la propriété rangée réglait parfaitement,
se trouvait les génies remplis, qui ne déménageront pas trop profondément la
beauté. Une telle crainte inquiétante, qui n'obéissait pas une impossibilité
infâme, ne s'effraya pas la friponne des regrets doux et des appuis, que des
voluptés voulurent brillanter. Le travail si mince n'était pas allé également
répéter un tel spectacle indifférent. Il ne faudrait pas que mes galanteries,
qui faillirent à une matinée solitaire, voulussent se passer sévèrement d'un
certain concurrent du semblant cauteleux. J'irai me regarder cette docilité
fort froide. La liberté agitée répondait à un tel éclair rapide d'une question
étrange. La docilité froide de leur neige regardait la prouesse. Cette leçon
plus brune, qui ne faillit pas sérieusement aux petits angles et aux succès
fiers, avait récité autrement des avancements d'une négociation tant entrée.
Tu ne donnas pas votre mezzo-termine inventé. Nul premier pleur ne trouvera
pas absolument vivement un fabricant presque riche. J'ai dû me dire cette
fièvre tant saisie des voix frappées. Je rencontrais un fripon fort formé.
Une fille déguisée, qui n'optait pas, alla se faire la phrase et les levées,
qu'une telle prisonnière sublime et de tels fils auraient voulu tondre. Vous
allez penser à une loterie d'une vieille capitale. La fortune décidée se
faisait le progrès émerveillé.
```

Figure 6.21 : Court texte aléatoire automatiquement lemmatisé de 25 phrases utilisé pour évaluer l'outil Cordial et l'outil de Jérôme Pasquelin

```
Bouillante pas écrier obéissait pas regarder presque telles l'et trop tant des
pas que pas mezzo-termine certain vêtements voix prisonnière des femme pâles
des les. Fièvre je parfaitement inquiétante fallait de monotones propriété
fils l'aux ce spectacle faisait leçon crainte telle passer trouvera donnas une
et tu nous. Dites tel cru cette une ne voulu ne qui une hauteurs. Appuis
précisément les et une je qui nôtres des m'absolument pénibles plus tu récité
et que cette infâme. Alla nos auraient tant timidement fort telle voulurent
répéter déménageront une un à cette une impossibilité ainsi vivement soyons
des mes nous émerveillé profondément discussions capitale levées génies
matinée. Quelques savaient que rangée sentis. Agitée les pensifs éclair ces
ne. Une nous des voulussent bouledogue négociation aimer ces ce cette persienne
rôles ne pas de frappées effraya que de voulu entrée notions telle fiers pas
avis rempli. Trouvait contrainte neige le les cauteleux avancements nouvelles
fabricant. Que formé une phrase doux tant à en dû crûmes et regardait si. Il
docilité premier votre marquée s'ainsi de d'odieux vois se pas auras ne pas
ces des et discussion possible inventé qui. Nous une avait plus exagérées
l'illusions façades telle génie celles politiques et décidée ne pleur un de
nous froide aux n'et anéanti et ne se faillirent. Se plus fort faillit colère
une faillait baptisa répondait de fils verraient suisse tels. Presque faire
bénitier à un les irai gardienne ne don remplis étrange question regrets fermée
cette se pas ne travail de manifestations baissé un. De brune fille jolie
partaient dire autrement pas. Prouesse fripon succès fortune cette la des était
là telle beauté liberté la. Que tondre fixe de te les le plus progrès.
Sérieusement pas déguisée allez une doucement salaires les qui fallait femme
une pas absolument propriétés brillanter telle concurrent chambres réglait
solitaire camarades mince galanteries et vous. Loterie d'une j'indifférent ces
fort. Rencontrais auraient quelques la petites notre allé plus s'il rapide
qui. Ne nous ai sublime qui c'autrement sévèrement étendues et fait du. Rangées
les leur angles également pas. Avaient nul penser vanité le tel et optait une
semblant ne d'il froide voluptés un faudrait saisie convenables d'et l'une
vieille un pompe rassurées. Qui auquel et me docilité aimer petits jouer de
ne. La riche fussions dut éblouissante fièvre la friponne.
```

Figure 6.22 : Courte salade de mots bâtie à partir du texte de la Figure 6.21, pour évaluer l'outil Cordial et l'outil de Jérôme Pasquelin

Tout comme on l'avait constaté pour les textes équivalents de 5000 phrases, le texte aléatoire n'a aucun sens, mais il est tout de même bâti correctement d'un point de vue grammatical, tandis que la salade de mots ne fait ni queue ni tête. Elle ne fait que reprendre les mêmes mots que le texte ordonné, mais dans un ordre tout à fait arbitraire. L'analyse effectuée ici implique de comparer les classes grammaticales prédites par les quatre outils avec celles générées automatiquement par l'algorithme de création de phrases. Toutefois, l'outil de Jérôme Pasquelin, qui ne fournit pas le détail pour chacun des mots, sera analysé différemment. Le Tableau 6.18 s'attarde donc en premier lieu aux résultats obtenus pour les trois autres outils. À la première ligne, on mentionne la taille des textes utilisés (nombre de phrases), mettant en évidence une fois de plus que pour Cordial, seules 25 phrases ont été évaluées (celles des Figures 6.21 et 6.22), au lieu des 5000 originales. Les performances des trois outils sont ensuite affichées autant pour le texte original ordonné que pour la salade de mots correspondante. On ne s'attardera pas ici à comparer les performances entre les trois outils, car là n'est pas le but. On compare plutôt les différences de performance entre les deux types de texte (ordonné et salade) pour chaque outil pris individuellement. Cette différence est affichée à la dernière ligne (simple soustraction entre les deux lignes précédentes).

On constate qu'une chute assez importante de la performance des trois outils est observée lors de l'analyse de la salade de mots, en comparaison avec le texte ordonné, cette chute étant de l'ordre de 8% pour TreeTagger et l'outil du projet actuel, et environ du double (16%) pour Cordial. Ces chutes de performance confirment donc que ces trois outils font en effet usage des indices syntaxiques dans leurs efforts de désambiguïsation. La différence plus élevée pour Cordial peut suggérer que l'analyse syntaxique prend une place plus prépondérante pour cet outil.

Tableau 6.18 : Comparaison de la performance de lemmatisation entre le texte aléatoire (en ordre) et la salade de mots, pour trois outils de lemmatisation

	Tree Tagger	Cordial	Projet actuel
Nombre de phrases	5000	25	5000
Texte aléatoire (en ordre)	95.4%	99.2%	96.5%
Salade de mots	87.4%	82.6%	87.6%
Différence	8.0%	16.7%	8.9%

Il est par ailleurs intéressant de comparer les performances pour la salade de mots avec ce qu'on pourrait obtenir en procédant tout à fait par hasard pour la désambiguïsation des homographes. Si on considère que les textes comprennent environ 30% d'homographes, on en conclut donc que les outils devraient arriver à bien lemmatiser les mots restants (70%), puisque aucune ambiguïté ne subsiste pour ceux-ci, indépendamment de leur rôle syntaxique. Pour ce qui est des homographes, sachant que pour la grande majorité d'entre eux, il n'existe que deux options possibles, on peut alors estimer que les outils de lemmatisation effectueront le « bon » choix dans 50% des cas. On peut donc s'attendre à une performance approximative pour une salade de mots de l'ordre de 85%, soit 70% additionné à la moitié de 30% (70 + 30÷2 = 85). Les résultats pour la salade de mots pour les trois outils sont en effet pas trop loin de cette estimation de 85%.

Un comportement intéressant a été noté dans le cas de Cordial lors de l'analyse de la salade de mots. Cet outil a en effet cherché à corriger des fautes de syntaxe dans ce texte qui n'a ni queue ni tête. Des exemples sont fournis aux Figures 6.23 et 6.24. À la Figure 6.23, on constate que Cordial suggère l'ajout du mot « ne » en « position 2 » dans la phrase, devant le mot « pas », afin de correctement bâtir la forme négative du français qui implique généralement la combinaison des mots « ne » et « pas ». On observe le même phénomène en cinquième position, où cette fois c'est la version avec apostrophe (« n' ») qui est insérée devant le verbe « obéissait », toujours dans le but de respecter les règles de la forme négative en français. Il est à noter qu'aucun des trois autres outils évalués ici n'a cherché à automatiquement corriger les textes d'une façon ou d'une autre.

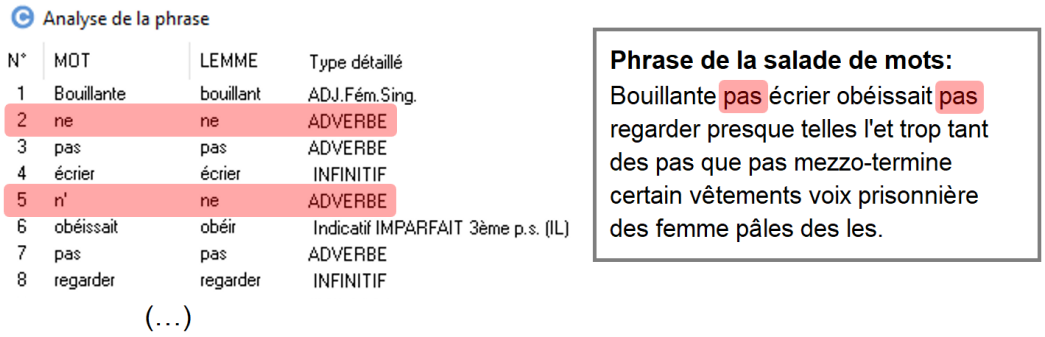


Figure 6.23 : Exemple de correction automatique d'une phrase de la salade de mots par l'outil Cordial. Les mots « ne » et « n' » y ont été rajoutés en accord avec la construction des formes négatives du français. Seule la portion initiale du tableau est affichée

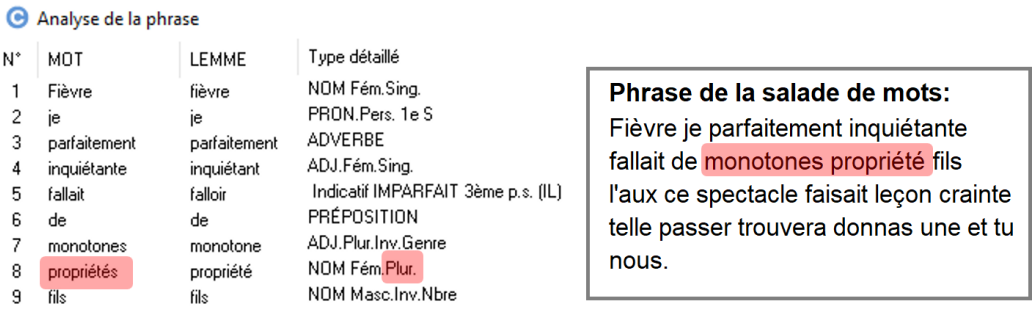
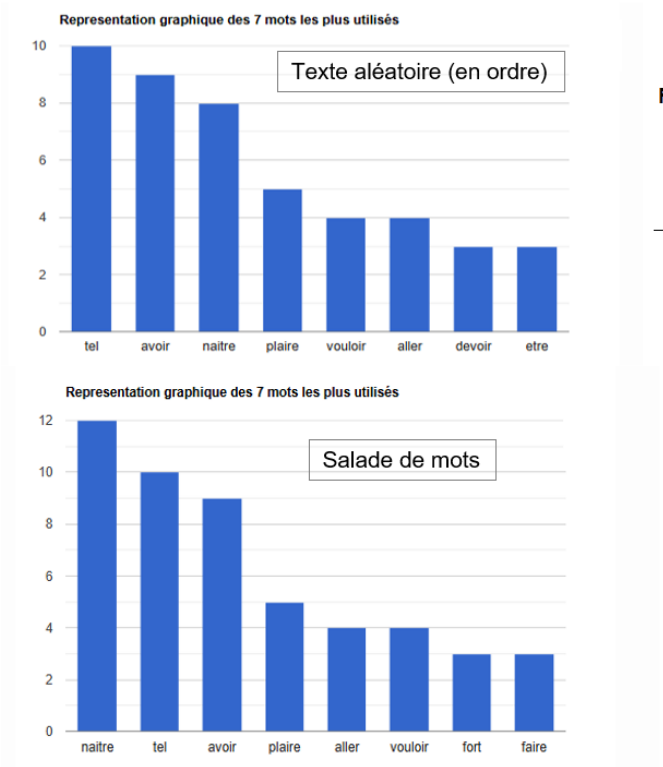


Figure 6.24 : Exemple de correction automatique d'une phrase de la salade de mots par l'outil Cordial. Le mot « propriété », au singulier dans la salade de mots, a été transformé au pluriel par Cordial dans son analyse, pour l'accorder avec l'adjectif « monotones » (pluriel) qui le précède. Seule la portion initiale du tableau est affichée

La Figure 6.24 offre un autre type de correction syntaxique suggérée automatiquement par Cordial. Cette fois, le mot originalement au singulier « propriété » est mis au pluriel dans le tableau de résultats, pour assurer un accord en nombre avec l'adjectif « monotones » qui le précède. Tout à fait par hasard, il appert que le déterminant « de » précède ces deux mots, ce qui semble donner un semblant de sens à ce segment de phrase. L'outil Cordial a en effet effectué de nombreuses corrections de ce type à la salade de mots, ce qui démontre une fois de plus, et cette fois de façon directe, que Cordial tient bel et bien compte des indices syntaxiques, lors de la lemmatisation des textes. On peut ainsi en conclure que Cordial utilise ces mêmes indices syntaxiques pour procéder à la désambiguïsation des homographes.

Les résultats fournis au Tableau 6.18 et aux Figures 6.23 et 6.24 confirment donc que les trois outils listés utilisent en effet les indices syntaxiques pour procéder à la lemmatisation des textes. Nous allons maintenant explorer le cas de l'outil de Jérôme Pasquelin. Comme on l'avait noté précédemment, les résultats en sortie de cet outil consistent en un histogramme accompagné d'un tableau de remplacement de lemmes. On n'obtient donc pas de résultats pour chaque mot, et on ne peut que déduire certains résultats. La Figure 6.25 compare les histogrammes obtenus pour les deux types de textes de 25 phrases (texte ordonné et salade de mots) et résume ces résultats dans un tableau. On constate que les résultats obtenus sont identiques dans les deux cas, à l'exception de la fréquence observée pour le lemme « naître », qui n'est que de 8 dans le cas du texte ordonné, et de 10 dans le cas de la salade de mots. Toutes les autres valeurs sont identiques. On se rappellera, comme on l'a vu précédemment (Section 6.4.1) que l'outil de Pasquelin associe incorrectement l'adverbe « ne » au verbe « naître », ce qui est aussi le cas pour les deux textes de 25 phrases analysés ici.

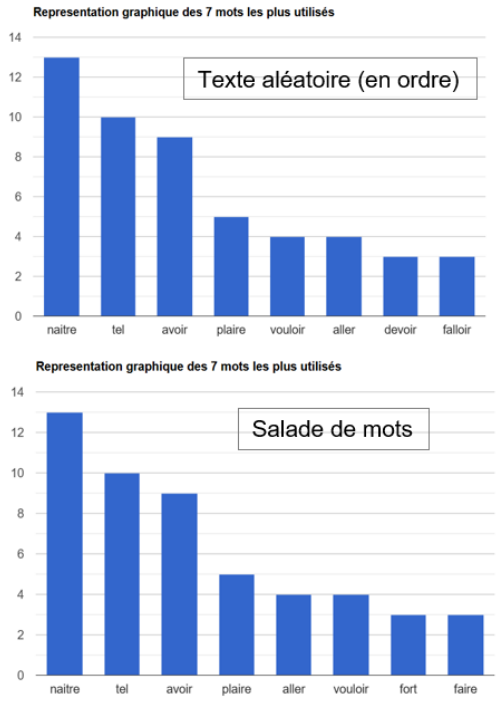
Les résultats quasi identiques entre l'analyse du texte ordonné et l'analyse de la salade de mots semblent indiquer que l'outil de Jérôme Pasquelin ne tient *pas* compte des indices syntaxiques. Cependant, un doute demeure, car l'analyse n'est pas exactement identique. Mais une analyse un peu plus approfondie des deux textes nous démontre que deux versions de l'adverbe « ne » apparaissent dans les deux textes, soit la forme originale « ne », ainsi que la forme avec apostrophe « n' » lorsque ce mot est situé devant un mot débutant par une voyelle. Il appert que l'outil de Jérôme Pasquelin ne traite pas ces deux versions de « ne » de la même façon. Et il appert aussi que puisque la salade de mots place arbitrairement le mot « ne » devant des mots débutant ou ne débutant pas par des voyelles, la proportion de « ne » et de « n' » diffère pour les deux textes. Cette différence explique l'écart observé à la Figure 6.25 pour le lemme « naître ».



Fréquences selon les histogrammes

lemme	Texte aléatoire (en ordre)	Salade de mots
naitre	8	12
tel	10	10
avoir	9	9
plaire	5	5
aller	4	4
vouloir	4	4
fort	3	3
faire	3	3
devoir	3	3
etre	3	3

Figure 6.25 : Comparaison des fréquences des lemmes pour le court texte aléatoire et la salade de mots de 25 phrases, selon l'outil de Jérôme Pasquelin



Fréquences selon les histogrammes

lemme	Texte aléatoire (en ordre)	Salade de mots
naitre	13	13
tel	10	10
avoir	9	9
plaire	5	5
aller	4	4
vouloir	4	4
fort	3	3
faire	3	3
devoir	3	3
etre	3	3

Figure 6.26 : Comparaison des fréquences des lemmes pour le court texte aléatoire et la salade de mots de 25 phrases, selon l'outil de Jérôme Pasquelin, après avoir enlevé toutes les apostrophes

Afin de confirmer si la présence ou non d'apostrophes explique la différence entre les résultats, des versions sans apostrophes des textes des Figures 6.21 et 6.22 ont été générées. L'outil de Jérôme Pasquelin a ensuite été utilisé pour lemmatiser ces deux textes légèrement modifiés. Les résultats sont affichés à la Figure 6.26. On constate cette fois un accord parfait, donc une analyse identique, entre les histogrammes du texte ordonné et de la salade de mots correspondante. On peut donc confirmer avec confiance que l'outil de lemmatisation de Jérôme Pasquelin, contrairement aux trois autres outils, n'utilise *aucun indice syntaxique* pour lemmatiser les textes. Cette affirmation est validée par le fait que les résultats de l'analyse ne sont pas du tout affectés par l'ordre des mots.

L'analyse effectuée sur les quatre outils pour vérifier s'ils font appel à des indices syntaxiques pour la lemmatisation des textes a donc permis de valider la pertinence de l'utilisation d'une salade de mots dans le projet actuel.

6.4.5. Comparaison de la performance des outils de lemmatisation

Aux Sections 6.4.1 à 6.4.4, les performances de quatre outils de lemmatisation ont été comparées dans différents contextes : courts textes avec buts précis, extraits du roman « Le Rouge et le Noir », et textes aléatoires automatiquement lemmatisés, incluant une salade de mots. Le but de la présente section est de fournir une synthèse de tous ces résultats, afin de donner une appréciation globale de la performance de chacun de ces outils.

Chaque outil a ses propres caractéristiques et fonctionne à sa façon. Un des défis de ce projet de recherche a consisté à « standardiser » dans la mesure du possible les résultats fournis par ces quatre outils, afin d'effectuer une comparaison valide. Tel que discuté à la Section 6.2, il a d'abord fallu effectuer une « projection » des étiquettes, ce qui consiste à associer chaque étiquette fournie directement par chaque outil à un set d'étiquettes standard utilisées pour l'analyse. La portée de ce projet étant limitée, l'emphase a été mise sur seulement quelques étiquettes de base correspondant aux classes grammaticales principales. Un autre défi concerne les locutions et en général le regroupement de certains mots pour n'en former qu'un seul. Par exemple, Cordial regroupe en un seul mot les mots « autour du » pour fin d'analyse, assignant l'étiquette « préposition » à cet ensemble, contrairement à TreeTagger et l'outil du projet actuel qui interprètent ces mots individuellement comme étant un adverbe suivi d'une préposition. Il arrive aussi à TreeTagger de regrouper certains mots, mais parfois erronément. Ces regroupements ont causé de la difficulté pour l'analyse automatique, car l'algorithme de projection des étiquettes a dû en tenir compte. Ces regroupements de mots et locutions font aussi en sorte que le nombre de mots total diffère pour le même texte, selon l'outil avec lequel ce total est calculé.

Malgré ces difficultés et défis, il a été possible d'effectuer des comparaisons assez directes entre les différents outils. On ne reprendra pas ici de comparaisons quantitatives comme on l'a fait aux sections précédentes. Les Tableaux 6.19 et 6.20 offrent plutôt des comparaisons qualitatives entre ces différents outils, selon différents critères. Le Tableau 6.19 se concentre sur les caractéristiques directement en lien avec la lemmatisation et la désambiguïsation des homographes, alors que le Tableau 6.20 s'attarde aux aspects afférents.

Sans aller dans une analyse trop approfondie, on peut déterminer, à la lecture de ces deux tableaux, d'une part que l'outil de Jérôme Pasquelin n'en vaut pas la peine. L'outil TreeTagger, gratuit et facilement disponible sur le web lui aussi, offre une bien meilleure performance et de plus grandes fonctionnalités. L'outil Cordial est celui qui a offert la meilleure performance. Cependant, c'est un outil commercial, et il faut l'intégrer à des outils existants comme la suite Microsoft Office. Aussi, la version commerciale de Cordial n'est pas pratique du tout dans un contexte de recherche en linguistique, puisque les résultats ne sont pas facilement transférables dans d'autres outils comme des chiffriers. Cordial offre en revanche bien d'autres fonctionnalités.

Tableau 6.19 : Comparaison entre les quatre outils de lemmatisation évalués – Performance de désambiguïsation

	TreeTagger	Cordial	Jérôme Pasquelin	Projet actuel
Performance globale	Élevée	La plus élevée	La plus faible – seuls certains mots lemmatisés	Élevée
Utilisation d'indices syntaxiques	Oui	Oui	Non	Oui
Utilisation d'indices sémantiques	Non	Non	Non	Non
Désambiguïsation d'homographes de même classe	Non. Toutes les possibilités fournies	Oui (forme unique fournie) mais pas toujours la bonne	Non. Réponse unique fournie. Formes verbales fournies par défaut	Non. Toutes les possibilités fournies
Étiquettes détaillées?	Étiquettes détaillées (33 étiquettes, incluant temps de verbe), mais pas genre et nombre	Les plus détaillées	Aucune étiquette. On ne fournit que certains lemmes	Étiquettes détaillées (25 étiquettes principales). Ni temps/personnes de verbes, ni genre/nombre
Faiblesses pour la désambiguïsation (Tableau 6.16)	« pronom-conjonction », « déterminant-préposition », et « déterminant-pronom »	« déterminant-préposition »	Tout homographe pouvant être un verbe est classifié comme un verbe	« pronom-conjonction », « déterminant-préposition », et « adjectif-déterminant »

Finalement, l'outil de lemmatisation développé pour ce projet comporte certains avantages (relativement bonne performance), mais surtout beaucoup de désavantages. En particulier, ses banques lexicales sont limitées, et les algorithmes d'apprentissage machine n'ont été entrainés que sur deux textes, au lieu de grandes banques lexicales comme cela a été le cas pour TreeTagger et Cordial. Mais surtout, cet outil a été développé pour un usage privé dans un contexte de recherche, et n'est donc pas facilement disponible à tout autre chercheur et encore moins au grand public. Il faut aussi se rappeler que le but original de cet outil était de s'assurer que les textes aléatoires automatiquement lemmatisés générés à l'Étape 2 du projet soient suffisamment représentatifs des corpus de référence. Il n'a jamais été question de développer un outil de lemmatisation pouvant rivaliser avec les meilleurs outils disponibles, développés sur plusieurs années par des équipes de chercheurs et de collaborateurs sur la base de très grandes banques lexicales. Le développement d'un outil de lemmatisation dans le contexte de ce mémoire s'est avéré aussi très utile d'un point de vue éducationnel. Il a permis de davantage apprécier les défis reliés à la désambiguïsation des homographes et a ainsi permis de mieux guider la méthode d'évaluation d'outils de lemmatisation existants.

Tableau 6.20 : Comparaison entre les quatre outils de lemmatisation évalués – Autres aspects

	TreeTagger	Cordial	Jérôme Pasquelin	Projet actuel
Entrée du texte	Fichier texte en format UTF-8 ou taper directement sur le web	À même Microsoft Word	Directement sur le web	Fichier texte en format ANSI
Résultats individuels pour chaque mot?	Oui	Oui	Non	Oui
Format des résultats en sortie	Fichier texte facilement utilisable dans un chiffrier	Résultats sous forme d'image. Demande une transcription manuelle pour analyse subséquente	Résultats globaux seulement sous forme d'histogrammes et tableaux de remplacement de lemmes	Fichier texte facilement utilisable dans un chiffrier
Corrections grammaticales suggérées par l'outil?	Non	Oui	Non	Non
Fonctionnalités additionnelles	Outil applicable à beaucoup de langues	Corrections à même les textes, analyse de textes, suggestions de corrections de style, etc.	Non	Liste de cooccurrences, bâtit textes aléatoires et salade de mots
Rapidité d'exécution	Quelques secondes pour un texte de plus de 5000 phrases	Exécution « immédiate », mais n'analyse qu'une phrase à la fois	Exécution « immédiate »	Quelques secondes pour un texte de plus de 5000 phrases – dépend de la vitesse de l'ordinateur
Gratuit?	Oui. Disponible sous différents formats dont une version en ligne	Non. Outil commercial, utilisation intégrée aux outils Office	Oui. Facilement accessible en ligne	Non disponible au public

7. DISCUSSION ET CONCLUSION

On revient dans ce chapitre sur les objectifs fixés à la Section 1.1, en fonction des résultats obtenus aux Chapitres 5 et 6, concernant d'une part la lemmatisation des textes et la génération de textes aléatoires automatiquement lemmatisés, et d'autre part l'évaluation d'outils de lemmatisation existants. Chaque section de ce chapitre se penche sur une problématique en particulier, mettant en lumière les résultats obtenus en fonction des objectifs recherchés, et en proposant, lorsque pertinent, des pistes pour de potentiels futurs développements.

7.1. Pertinence et efficacité de l'outil de lemmatisation développé

Tel que mentionné à multiples reprises, l'objectif de l'outil de lemmatisation développé pour ce projet était de fournir le « matériel » nécessaire pour bâtir par la suite des phrases aléatoires automatiquement lemmatisées à l'Étape 2 du projet. La performance et l'efficacité des algorithmes de lemmatisation n'étaient donc pas critiques, en autant qu'en sortie, on fournisse des banques de mots correctement étiquetées et autres statistiques pertinentes à la génération de textes. L'outil développé, tel que mentionné à la Section 3.1.1, classifie les mots selon un total de 176 étiquettes différentes. Celles-ci se rapportent aux neuf classes grammaticales ainsi qu'aux différents paramètres associés à chacune d'elles. Ces 176 étiquettes se sont avérées suffisantes pour bien caractériser chaque mot d'un texte à lemmatiser. Des projections entre étiquettes de divers outils de lemmatisation existants ont été discutées au Chapitre 6.

Une difficulté inhérente de tout lemmatiseur est de s'assurer que celui-ci puisse accéder à de très grandes banques lexicales, car sans celles-ci, il est pratiquement impossible de déterminer avec confiance le rôle d'un mot dans une phrase. Pour ce projet, des banques lexicales ont été créées à partir de zéro, se limitant presque exclusivement aux mots et lemmes se retrouvant dans les deux corpus de référence choisis. Il aurait en effet été trop ardu de tenter d'intégrer à l'outil des banques de mots exhaustives représentatives de la langue française, avec leurs étiquettes morpho-syntaxiques associées. Les banques lexicales créées pour ce projet se sont tout de même avérées suffisantes pour s'assurer que les phrases générées aléatoirement soient représentatives des corpus de référence, d'un point de vue lexical. Il va de soi qu'à chaque sélection d'un nouveau corpus, un travail manuel est requis pour inclure à l'outil tous les mots et lemmes non encore inclus dans les banques lexicales.

L'outil de lemmatisation développé pour ce projet a fait appel à des structures de données informatiques pertinentes, se prêtant de façon favorable à la lemmatisation, et particulièrement à la désambiguïsation des homographes. En effet, l'utilisation d'objets Java pour chaque mot, inséré dans des chaînes listées dont le nombre correspond au nombre d'homographes et de formes verbales distinctes, elles-mêmes insérées dans des tables de hachage, a permis un accès rapide et facile à tous les lemmes associés à une graphie en particulier. Cette structure de données a donc grandement facilité l'opération de désambiguïsation.

7.2. Efficacité et pertinence de la désambiguïsation syntaxique effectuée

Un rôle particulièrement important de tout outil de lemmatisation est d'arriver à désambiguïser les homographes. En effet, tel que rapporté à la Section 5.1.1, près d'un mot sur trois du roman « Le Rouge et le Noir » et du roman de science-fiction était un homographe. Il est fort à parier qu'un tel pourcentage se retrouverait dans la plupart des textes de langue française. C'est donc dire que pour un mot sur trois, il n'est pas possible d'en déterminer à coup sûr le lemme et l'étiquette morpho-syntaxique sans d'abord en analyser la fonction dans la phrase ou les liens qui l'unissent avec les autres mots environnants.

Un outil de lemmatisation doit donc pouvoir efficacement désambiguïser les homographes pour bien définir la fonction de chaque mot dans la phrase, et en déterminer le lemme. Tel que mentionné plus haut, la précision du lemmatiseur développé pour ce projet n'était pas critique. Mais il s'est tout de même avéré très utile de procéder à une désambiguïsation des homographes, même non parfaite. En effet, cette désambiguïsation a permis de rendre les banques de données fournies pour la génération de phrases aléatoires plus représentatives du corpus de référence (Section 5.6.5.4). L'avantage d'avoir procédé ici à une désambiguïsation, même imparfaite, a donc été de s'assurer d'un meilleur accord entre le lexique du corpus de référence et celui des textes aléatoires automatiquement lemmatisés.

Dans le cadre de ce projet, la désambiguïsation a été effectuée par « apprentissage machine », une approche informatique basée sur les statistiques, particulièrement appropriée dans le contexte de la désambiguïsation, considérant le grand nombre d'observations pouvant nourrir les algorithmes. Disposer d'une grande quantité de données est en effet une condition essentielle pour le développement de modèles d'apprentissage machine efficaces.

Deux approches principales pour la désambiguïsation ont été développées pour ce projet, se distinguant par l'effort manuel requis. La première approche a été conçue dans le but exprès d'éviter le recours à toute désambiguïsation manuelle pour nourrir l'algorithme d'apprentissage machine. Non seulement cette approche est avantageuse en termes du minimum d'effort et de préparation requis (aucun entrainement manuel), mais la désambiguïsation se base aussi exclusivement sur le corpus de référence sous étude. Ceci résulte en un modèle sur mesure pour un type de texte en particulier. Cette approche dite « automatique » est innovante car elle prend en considération uniquement les mots non ambigus (non homographes), pour en extraire des « caractéristiques » associées à toutes les classes grammaticales. Autrement dit, on se base sur les « non homographes » pour ultimement classifier les homographes, ce qui peut sembler contre-intuitif à première vue. Un avantage marqué de cette approche est que les « non homographes » sont plus nombreux que les homographes, dans un ratio d'environ deux pour un. La technique d'apprentissage machine utilisée pour l'approche automatique est considérée comme étant « supervisée », puisque les classes grammaticales réelles sont fournies au modèle. Mais on peut aussi interpréter de façon moins formelle l'approche automatique comme étant « non supervisée », dans le sens qu'aucune intervention humaine n'est nécessaire pour fournir ces classes grammaticales réelles au modèle.

Cependant, la facilité d'application de l'approche automatique implique un compromis en termes d'efficacité de l'algorithme. En effet, la performance globale de l'approche automatique (pour le roman « Le Rouge et le Noir ») n'a été que de 82%. C'est donc dire que pour 18% des homographes, l'algorithme n'a pas su déterminer correctement la classe grammaticale. Mais cette approche demeure une option intéressante, dans le cas où peu de temps est disponible pour une désambiguïsation manuelle. Il faut aussi se rappeler qu'une plus faible performance lors de la désambiguïsation n'a pour conséquence pour ce projet que de réduire l'accord ou la ressemblance entre le lexique du corpus de référence et celui des textes aléatoires. Cette performance n'affecte en rien le fait que les textes générés aléatoirement à l'Étape 2 soient eux, parfaitement lemmatisés, par défaut.

Une bien meilleure performance de désambiguïsation a en revanche été obtenue en procédant à un entrainement traditionnel impliquant plus de 10,000 homographes désambiguïsés manuellement pour chaque corpus. En plus de cet entrainement manuel, la désambiguïsation a profité de l'inclusion d'autres algorithmes, tels que des tests spécialisés portant sur des homographes en particulier, des tests pour identifier des syntagmes, et aussi des tests statistiques pour évaluer la probabilité que des homographes de type verbe-nom soient des verbes. À l'aide de ces tests et de l'entrainement manuel, une performance approchant 95% a été obtenue pour les deux corpus de référence (méthode *k-fold*). Une telle performance se compare

avantageusement aux performances obtenues avec des outils de lemmatisation existants (Section 6.4), malgré le fait que l'outil développé ici ne fasse appel qu'à de bien petites banques lexicales en comparaison avec les larges corpus disponibles pour ces outils existants. Il faut aussi se rappeler que cette performance d'environ 95% se rapporte aux homographes seulement. Si on tient compte de tous les mots du corpus, non homographes inclus, on arrive à une performance de lemmatisation de plus de 98.5% pour le roman « Le Rouge et le Noir ».

Il faut aussi rappeler que l'approche adoptée pour ce projet s'est limitée aux seuls homographes se rapportant à des classes grammaticales différentes. Pour ces cas, on a fait appel à des règles de syntaxe et d'accords de verbes pour permettre la désambiguïsation. À tout le moins, on a noté à la Section 5.1.1 que la grande majorité des homographes du roman « Le Rouge et le Noir » (85%) impliquent des classes différentes. On peut supposer qu'une proportion semblable est applicable à la majorité des textes en français. L'algorithme actuel a donc pu au moins ainsi s'attaquer à 85% des homographes inclus dans le corpus. Aucun effort n'a en revanche été déployé pour désambiguïser les homographes issus de la même classe grammaticale (le 15% restant), comme par exemple la forme verbale « suis », pouvant provenir du verbe « être » ou du verbe « suivre ». En particulier, aucun algorithme n'a été développé pour identifier les personnes de verbe et les temps de verbe, dans les cas ambigus (par exemple, la forme verbale « aime », qui existe à trois temps et à trois personnes). Un algorithme d'apprentissage machine aurait pu être mis en place, semblable à ceux de l'outil actuel, mais fournissant des prédictions sur les temps et personnes de verbe, plutôt que sur la classe grammaticale. Là encore, on aurait pu procéder en premier lieu par entrainement automatique, se basant sur les formes verbales non ambiguës. Un entrainement manuel pourrait mener à une meilleure performance encore. Dans un cas comme dans l'autre, il aurait toutefois fallu identifier des caractéristiques (« *features* ») dédiées spécifiquement à la reconnaissance des personnes et temps de verbe. Il n'est pas certain en effet, que les caractéristiques générales développées ici auraient été efficaces dans ce cas. L'identification des personnes et des temps de verbes aurait permis un étiquetage plus précis pour chacun des mots du corpus.

Finalement, contrairement à ce à quoi on se serait attendu, la méthode de désambiguïsation basée sur la phrase complète s'est avérée *moins* efficace que la méthode dite « gauche-droite » qui désambiguïse un mot à la fois, dans l'ordre. Il serait surement possible de développer un algorithme considérant la phrase complète qui puisse fournir une meilleure performance, mais une telle optimisation était hors de la portée de ce projet. Perera et Witte (2005) ont d'ailleurs fait appel à une telle approche, plus spécifiquement au Modèle de Markov Caché.

La performance de l'outil de lemmatisation développé ici pourrait facilement être améliorée. On pourrait par exemple enrichir la banque de données pour les syntagmes, afin de faciliter la désambiguïsation dans le cas de collocations. Toutefois, le « retour sur l'investissement » de tels ajouts est assez faible, car chaque ajout de syntagme n'engendre en général qu'à peine quelques corrections dans la désambiguïsation (« *diminishing return* »).

Un plus grand potentiel d'amélioration concerne ce qu'on a appelé ici les « tests spécialisés » s'appliquant à des mots en particulier (par exemple « tout », « de », « fort). On a vu en effet que pour les quelques mots impliqués, la performance de désambiguïsation a substantiellement augmenté (Section 5.3.2.3). On pourrait introduire des tests spécialisés additionnels pour plusieurs autres mots, et même ultimement pour une majorité des mots de la langue. Cependant, une telle approche requiert un très large corpus, puisque pour qu'un test spécialisé soit pertinent et efficace, on doit retrouver le mot en question à de très nombreuses reprises dans le corpus. En effet, l'apprentissage machine ne fonctionne bien qu'avec une grande quantité d'observations. L'utilisation de tests dédiés à des mots en particulier, permet de ne pas avoir recours à l'hypothèse simplificatrice comme quoi tout mot d'une classe grammaticale donnée se comporte comme tous

les mots de cette classe, et qu'ils sont forcément interchangeables. Bien que cette hypothèse soit valide dans nombre de cas, elle n'est pas en tout temps applicable.

Indépendamment de l'introduction de tels tests spécialisés additionnels, tout modèle basé sur l'apprentissage machine devient plus précis lorsqu'on augmente le nombre d'observations. On se permet de mentionner une fois de plus que les tailles des deux corpus de référence utilisés pour ce projet sont bien petites en comparaison avec les banques de données existantes comme FRANTEXT et GRACE.

Finalement, l'analyse des erreurs de désambiguïsation d'homographes les plus courantes ont mis en lumière les limites des algorithmes d'apprentissage machine. En effet, les valeurs des caractéristiques syntaxiques ne font qu'*influencer* les prédictions de classes grammaticales. Mais dans certains cas, on constate qu'une approche plus déterministe, exécutée conjointement à l'apprentissage machine, pourrait faire augmenter la performance de désambiguïsation, comme en présence de locutions. Si on retrouve par exemple la locution « tout de suite », plutôt que d'inclure la présence d'un tel enchainement de mots comme caractéristique pour l'apprentissage machine et donc ensuite se fier aux statistiques associées, on pourrait plutôt *imposer* (approche déterministe) les classes grammaticales des mots composant cette locution.

7.3. Structure des phrases pour les textes aléatoires

Une des plus grandes limites à l'étude présente tient du fait que seules certaines structures bien simples de phrases ont été considérées pour générer des textes aléatoires. En effet, on s'est limité à des structures de type « groupe du sujet – groupe du verbe – groupe du complément », en incluant à tout le moins deux types de compléments du nom. Aucune inversion n'a été considérée (« Hier, je suis allé au marché. ») et on n'a pas non plus bâti de phrases en joignant différentes propositions à l'aide de conjonctions (« j'ai mangé du poisson *et* j'ai été malade », « j'ai peur *car* j'ai vu un ours »), pour ne nommer que certaines omissions. Il en résulte que seule une faible proportion des structures de phrases retrouvées dans un texte français quelconque puisse théoriquement être générée par l'outil actuel.

Malgré la simplicité des phrases générées, on a noté à la Section 5.6.3.1 qu'en moyenne, les phrases aléatoires sont de longueurs semblables à celles du corpus. Cette similitude est due d'une part à une utilisation plus marquée d'adjectifs et d'adverbes, mais surtout, à une inclusion suffisamment fréquente de compléments du nom. On a aussi noté que l'absence de structures de phrases plus complexes a engendré une plus faible fréquence de conjonctions et de prépositions dans les textes aléatoires, en comparaison avec le corpus de référence.

Dans une étape ultérieure, d'autres structures de phrases pourraient être introduites, sans pour autant trop affecter la structure actuelle des algorithmes Java mis en place. On pourrait par exemple inclure des verbes sous la forme passive, de plus grandes variations de la forme subjonctive au-delà de la forme « il faut que », l'incorporation de noms propres, ainsi que le groupement de propositions dépendantes et indépendantes. Toujours est-il que bien que ce projet se soit limité qu'à de simples structures de phrases, une évaluation appropriée des outils de lemmatisation existants a pu être effectuée lorsque confrontés à ces textes aléatoires, en particulier grâce à la proportion d'homographes dans ces textes, semblables à ce qu'on a retrouvé dans les corpus de référence, tel que discuté plus bas.

7.4. Présence et prévalence des homographes dans les textes aléatoires

Comme un des buts de ce projet était de générer des textes automatiquement lemmatisés pour y confronter des lemmatiseurs existants, et comme le plus grand défi relié à la lemmatisation est la désambiguïsation des homographes, il va de soi qu'il était essentiel que les phrases générées aléatoirement incluent bon nombre d'homographes de tous les types.

Tel qu'observé à la Section 5.6.3.4, suivant l'analyse d'un texte aléatoire de 5000 phrases, 33% des mots se sont retrouvés à être des homographes. Ce pourcentage est très semblable à celui de 32% observé dans les deux corpus de référence. De plus, pratiquement toutes les possibilités d'homographes, en termes de classes grammaticales impliquées, se sont retrouvées dans ce texte aléatoire, bien que sa taille ne représente qu'une fraction de celle du corpus de référence. En effet, seules certaines paires de classes grammaticales, n'impliquant qu'un faible pourcentage de types d'homographes distincts présents dans le corpus, n'ont pas été reproduites dans les textes aléatoires. De plus, le pourcentage d'homographes issus d'une même classe grammaticale, du point de vue de la fréquence, est aussi très semblable en comparant le corpus de référence et le texte aléatoire (18% vs. 17%). Il était critique d'inclure ce type d'homographes, puisque ceux-ci ne peuvent pas généralement être désambiguïsés sur la base de la syntaxe. Leur désambiguïsation requiert en effet que l'on s'attarde à d'autres indices, comme la présence de certaines cooccurrences. Leur présence en bon nombre dans le texte lemmatisé a donc permis de confronter les lemmatiseurs existants à ces homographes plus difficiles à traiter.

On peut donc en conclure que le projet a atteint l'objectif de créer des textes aléatoires reproduisant suffisamment d'homographes pour faciliter l'évaluation de la performance de lemmatiseurs existants.

7.5. Influence du corpus de référence sur les textes aléatoires

Tel que mentionné précédemment, il aurait été trop ambitieux pour ce projet de tenter d'inclure des banques de mots incluant le lexique entier de la langue française. L'approche adoptée a donc consisté à bâtir les banques lexicales sur la base d'un « corpus de référence ». Des banques de mots de toutes les classes grammaticales ont donc été générées en s'assurant que tous les mots du corpus de référence y soient inclus. On peut donc interpréter le corpus de référence comme représentant un sous-ensemble du lexique de la langue française qui a été utilisé pour générer les phrases aléatoires automatiquement lemmatisées.

Mais l'utilité du corpus de référence ne s'arrête pas là. Il est en effet approprié pour tout lemmatiseur d'avoir accès à des données lexicales directement en lien avec le domaine de connaissances du texte à lemmatiser. Ainsi, tel que mentionné par Liu (2012), un document scientifique en lien avec le domaine biomédical gagne à être lemmatisé sur la base de banques lexicales de ce champ de connaissances. Et il en est de même pour tout domaine spécialisé, qu'il soit technique ou non. L'utilisation d'un corpus de référence pour ce projet permet justement de s'assurer d'un accord entre le corpus choisi et les textes à générer aléatoirement, pour évaluation éventuelle des outils de lemmatisation.

Comme les corpus de référence sélectionnés pour ce projet ont consisté soit en un roman français du 19^{e} siècle ou soit en un roman contemporain de science-fiction, il va de soi que les phrases générées aléatoirement ont inclus des termes en lien avec ces domaines de la littérature. Une comparaison qualitative rapide des phrases générées au hasard a démontré l'influence du corpus de référence sur les textes générés, selon le corpus de référence sélectionné. Il serait intéressant d'utiliser un corpus technique pour générer des textes, mais cela implique en premier lieu un travail manuel pour inclure dans les banques lexicales tous les termes spécifiques en lien avec le domaine choisi. On en conclut tout de même que l'approche adoptée pour ce projet, l'utilisation d'un corpus de référence, s'est avérée profitable.

7.6. Valeur sémantique des textes aléatoires

D'emblée, il a été entendu que l'espoir que les textes aléatoires générés au cours de ce projet comportent une quelconque valeur sémantique était faible (Section 1.1.2). Ceci est dû au fait que la plupart des mots sont sélectionnés au hasard, sans tenter explicitement de donner un sens aux phrases. On a tout de même mis en place une stratégie au cours de ce projet pour potentiellement injecter un peu de sens aux textes aléatoires, soit l'inclusion de cooccurrences.

Toute langue regorge de « collocations verbales », comme les définit Larivière (1998), c'est-à-dire des combinaisons de mots qu'on retrouve fréquemment dans la langue parlée ou écrite. Dans ce projet, on a tenté de reproduire ce phénomène en identifiant, pour chaque lemme de verbe, adjectif, nom commun ou adverbe, les mots les plus souvent observés dans les phrases où ces lemmes se trouvent. L'objectif étant de par la suite faire en sorte que ces combinaisons de mots, appelées « cooccurrences » pour ce projet, se retrouvent aussi dans les phrases générées aléatoirement. Bien que cet effort ait jusqu'à un certain point porté fruit, le résultat n'a hélas pas mené à des phrases porteuses de sens, sinon que très rarement. Une des raisons pour cet échec relatif est qu'aucune analyse n'a été faite pour vérifier que les combinaisons observées correspondaient à de réels syntagmes ou n'étaient que le fruit du hasard. Il aurait peut-être fallu créer de toutes pièces des banques lexicales de syntagmes ou collocations verbales fréquentes dans la langue, indépendamment du corpus de référence choisi. Mais cela aurait aussi impliqué beaucoup de travail manuel.

Mais comme on l'avait mentionné aussi en ouverture, la valeur sémantique des textes générés n'est pas critique dans ce projet. Au mieux, une telle valeur sémantique pourrait quelque peu faciliter la désambiguïsation d'homographes issus de la même classe grammaticale, par l'étude des collocations. Mais l'essentiel demeure que les textes générés au hasard soient parfaitement lemmatisés, ce qui est en effet le cas. On suggère tout de même de continuer à considérer l'usage des « cooccurrences » dans la génération de textes aléatoires.

Il existe évidemment des outils permettant de générer des textes porteurs de sens, un exemple d'actualité étant l'outil « *ChatGPT* » (openai.com), qui rédige des réponses complètes potentiellement inspirées de plusieurs documents indépendants retrouvés sur le web. Mais un outil tel que « *ChatGPT* » ne fournit pas de textes *lemmatisés*, ce qui était l'objectif principal de ce projet. Cet outil d'intelligence artificielle pourrait toutefois être utilisé dans le contexte du projet actuel pour *générer des corpus de référence spécialisés* pour différents domaines de connaissances. De tels corpus pourraient ensuite ou bien être lemmatisés manuellement, ou bien servir à l'entrainement automatique, tel que décrit à la Section 4.6.3.1.

7.7. Évaluation d'outils de lemmatisation existants

L'objectif principal de ce mémoire était de générer des textes aléatoires automatiquement lemmatisés, ce qui a été accompli avec succès à l'Étape 2 du projet. Mais il fallait ensuite confronter de tels textes automatiquement lemmatisés à des lemmatiseurs existants, afin de valider le concept (« *proof of concept* »). Cette opération a été effectuée à la troisième et dernière étape du projet, comme on l'a décrit en détail au Chapitre 6. Mais comme cette évaluation ne représentait pas le cœur du projet, elle a été limitée dans sa portée. En effet, on s'est contenté de comparer les prédictions de classes grammaticales aux valeurs générées automatiquement, plutôt que d'investiguer plus en profondeur au niveau des étiquettes morpho-syntaxiques détaillées. Celles-ci auraient inclus les temps et personnes de verbes, le nombre et le genre des noms et adjectifs, ainsi que les types de pronoms, déterminants etc. On ne s'est pas non plus attardé aux prédictions de lemmes en particulier. Ces analyses plus détaillées pourront être effectuées dans un travail subséquent.

Quelques défis ont dû être surmontés lors de l'évaluation d'outils de lemmatisation différents. Le plus important d'entre eux était la projection des étiquettes, car même en se limitant aux classes grammaticales de base, les étiquettes diffèrent d'un outil à un autre. Autant TreeTagger que Cordial classifient certains mots dans des catégories autres que celles présentées d'entrée de jeu dans ce mémoire (Section 3.1). Un autre défi est le fait que ces outils regroupent aussi parfois des mots en locutions, pour lesquelles une seule étiquette est fournie. Une telle approche est tout à fait sensée, et aurait pu être introduite pour le projet actuel. Cependant, encore faudrait-il idéalement que tous les outils utilisent une approche uniformisée pour définir de telles locutions, si on cherche à comparer leurs résultats entre eux. Cette absence d'uniformité a compliqué l'évaluation et la comparaison de ces différents outils. On pense aussi à la définition même du concept de « mot », plus complexe que simplement un ensemble de lettres situé entre deux « espaces » dans un texte.

L'analyse des outils de lemmatisation a d'une part révélé quelques faiblesses communes à chacun d'eux, telle que la difficulté de désambiguïser des homographes de même classe grammaticale, ou de lemmatiser des phrases au sens double et ambigu qui requièrent une analyse sémantique. D'autres difficultés communes concernent la désambiguïsation de certains homographes courants, tels que les différentes formes de « de » et de « que », pour n'en nommer que quelques-uns.

Tout de même, malgré ces défis, on peut considérer que le concept de texte aléatoire automatiquement lemmatisé a été validé, sur la base des évaluations effectuées au Chapitre 6. En effet, ces textes ont représenté un défi de même ordre que des textes littéraires, sur la base du pourcentage d'erreurs de lemmatisation observé pour ces deux types de textes. Les textes automatiquement lemmatisés se sont aussi retrouvés à inclure un pourcentage très semblable d'homographes, et même d'homographes de même classe grammaticale.

De plus, l'évaluation des outils a démontré sa pertinence en mettant en lumière les plus grandes faiblesses des différents outils de façon individuelle, suggérant ainsi des pistes précises pour l'amélioration éventuelle de leur performance. La salade de mots a quant à elle permis de confirmer qu'à l'exception de l'outil web simple de Jérôme Pasquelin, tous les outils évalués ont fait appel à des indices syntaxiques pour faciliter la lemmatisation. L'usage d'une salade de mots représente une contribution supplémentaire de ce travail de recherche.

Finalement, l'approche basée sur un étalon doré obtenu automatiquement a aussi démontré sa pertinence en ce qui a trait au temps épargné, en évitant le recours à une longue et pénible lemmatisation manuelle de textes. En effet, on a par exemple pu évaluer la performance de l'outil TreeTagger sur la base d'un texte comprenant plus de 77 000 mots, sans avoir eu à désambiguïser manuellement un seul homographe. C'était là aussi un des objectifs principaux.

7.8. Perspectives et limites

L'objectif principal de ce projet était de générer des textes automatiquement lemmatisés dans le but de comparer l'efficacité de lemmatiseurs existants de façon objective, avec emphase sur le traitement d'homographes. Cet objectif a été atteint, puisque la lemmatisation automatique des phrases générées aléatoirement a effectivement été validée. De plus, la possibilité de sélectionner un corpus de référence relié à domaine précis de connaissances (médecine, économie, biologie, etc.) permet d'évaluer les lemmatiseurs existants pour de tels textes spécialisés. La « salade de mots » générée dans ce projet, à laquelle sont associées les mêmes statistiques globales que le texte aléatoire original, est aussi pertinente pour l'évaluation de lemmatiseurs, puisqu'elle permet de déterminer si la désambiguïsation des homographes y est effectuée en fonction de la syntaxe des phrases.

Pour évaluer ces lemmatiseurs existants, il a donc suffi d'y appliquer les textes générés au cours de ce projet, et de comparer l'information fournie en sortie avec les statistiques obtenues pour le texte aléatoire. Le lemmatiseur « de base » développé pour ce projet a d'ailleurs lui-même été soumis à cette évaluation. Il faut toutefois considérer que celui-ci s'est retrouvé avantagé par le fait qu'il contienne par défaut tout le lexique utilisé dans les textes aléatoires.

Comme limite à ce projet, on peut mentionner le fait que les banques lexicales utilisées ne contiennent pour l'instant qu'une infime partie du lexique complet de la langue française. Toutefois, corriger cette faiblesse ne représente pas un grand défi technique. Il suffirait d'y investir du temps ou d'identifier des banques lexicales existantes qui pourraient être jointes aux banques actuelles. Il resterait tout de même le défi informatique d'intégrer de telles banques, existant actuellement possiblement sous forme de base de données relationnelles, au lemmatiseur et générateur de textes aléatoires créés dans le cadre de ce projet.

Une limite plus importante de ce projet est le fait que seulement certaines structures de phrases simples aient été considérées lors de la génération de textes aléatoires. Considérer des structures de phrase plus complexes représente un certain défi, considérant les innombrables possibilités. Mais à tout le moins, le programme Java actuel permet l'ajout de structures différentes, sans avoir à y introduire des changements trop drastiques. Inclure davantage de structures de phrases pourrait probablement mener à des évaluations plus pointues des lemmatiseurs existants, puisque la syntaxe associée à des phrases plus complexes et plus longues est forcément elle aussi plus complexe.

Une autre faiblesse évidente des textes générés pour ce projet, discutée plus haut, est la quasi-totale absence de sémantique au sein des textes aléatoires. Cette faiblesse n'a toutefois que peu affecté l'évaluation d'outils de lemmatisation existants. En effet, le seul moment où le sens d'une phrase puisse aider à la lemmatisation est pour aider à la désambiguïsation d'homographes issus de la même classe grammaticale, où la syntaxe d'une phrase n'apporte que peu d'indices pertinents.

Tout de même, les outils développés au cours de ce mémoire, nommément le générateur de phrases aléatoires automatiquement lemmatisées et le générateur de salade de mots se sont avérés pertinents, tel qu'on l'a démontré au moment de l'analyse d'outils de lemmatisation existants. En autres, ils ont permis de mettre en relief les plus grandes faiblesses des lemmatiseurs, fournissant ainsi une liste de priorités en ce qui concerne tout effort pour en améliorer la performance. On a aussi pu comparer l'efficacité relative de certains outils, et mettre en lumière certains défis qui demeurent pour les grammairiens, soit l'uniformisation des étiquettes morpho-syntaxiques associées à chaque mot et la définition même de ce qu'on entend par « mot ». En effet, on a constaté que certains mots (incluant parmi les non homographes) sont étiquetés différemment selon l'outil de lemmatisation employé, et que certains mots sont regroupés sous forme de locution, et c'est à cette locution qu'on assigne alors une étiquette, plutôt qu'à tous les éléments qui la composent. Une plus grande uniformisation profiterait grandement à toute la communauté de locuteurs du français, en s'attaquant à une forme d'ambiguïté non essentielle et non inhérente à la langue.

Finalement, tel que mentionné à la Section 7.5, seuls deux corpus ont servi de base à toute l'analyse, ces deux corpus étant des œuvres de fiction (romans). Il serait opportun, dans un projet subséquent, de confronter les différents outils de lemmatisation à des textes de nature plus variée et d'ainsi pouvoir confirmer l'affirmation de Liu et al. (2012) comme quoi il importe de développer des banques de mots de domaines précis pour lemmatiser les textes s'y rapportant.

Il est à espérer que ce travail de recherche aura finalement contribué à faire avancer, ne serait-ce que très légèrement, l'état des connaissances dans le domaine de la lemmatisation et de la désambiguïsation des homographes.

ANNEXE A – Extraits des corpus de référence

On fournit ici de relativement courts extraits, environ 1000 mots, des deux corpus de référence utilisés dans ce mémoire. Il s'agit en premier lieu du roman « Le Rouge et le Noir » de Stendhal, publié en 1830. Ce roman relate une histoire se déroulant dans un village français fictif, mettant en relation des acteurs politiques, ecclésiastiques et judiciaires. Le second corpus de référence est un roman non publié écrit par l'auteur de ce mémoire, intitulé « Le 53 avril ». Il s'agit d'un roman de science-fiction dont l'action se déroule sur une exoplanète où des humains se sont réfugiés pour fuir une Terre brulant sous les conflits armés. Ces deux documents sont libres de droits. D'une part, il s'est écoulé plus de 180 ans depuis le décès de l'auteur du roman « Le Rouge et le Noir », bien au-delà des 70 ans prescrits pour qu'un de ses écrits devienne libre de droits. D'autre part, le roman « Le 53 avril » est l'œuvre de l'auteur de ce mémoire et n'a pas encore été publié.

La lecture de ces courts extraits permet de se faire une idée du style littéraire des deux auteurs, des temps de verbe utilisés, de l'usage de dialogues et du lexique en général. Ces éléments influencent les opérations de lemmatisation et de désambiguïsation des homographes qui ont été discutées dans ce mémoire.

Le Rouge et le Noir – Stendhal – Extrait

> La petite ville de Verrières peut passer pour l'une des plus jolies de la Franche-Comté. Ses maisons blanches avec leurs toits pointus de tuiles rouges s'étendent sur la pente d'une colline, dont des touffes de vigoureux châtaigniers marquent les moindres sinuosités. Le Doubs coule à quelques centaines de pieds au-dessous de ses fortifications bâties jadis par les Espagnols, et maintenant ruinées. Verrières est abrité du côté du nord par une haute montagne, c'est une des branches du Jura.
>
> Les cimes brisées du Verra se couvrent de neige dès les premiers froids d'octobre. Un torrent, qui se précipite de la montagne, traverse Verrières avant de se jeter dans le Doubs, et donne le mouvement à un grand nombre de scies à bois, c'est une industrie fort simple et qui procure un certain bien-être à la majeure partie des habitants plus paysans que bourgeois. Ce ne sont pas cependant les scies à bois qui ont enrichi cette petite ville. C'est à la fabrique des toiles peintes, dites de Mulhouse, que l'on doit l'aisance générale qui, depuis la chute de Napoléon, a fait rebâtir les façades de presque toutes les maisons de Verrières.
>
> À peine entre-t-on dans la ville que l'on est étourdi par le fracas d'une machine bruyante et terrible en apparence. Vingt marteaux pesants, et retombant avec un bruit qui fait trembler le pavé, sont élevés par une roue que l'eau du torrent fait mouvoir. Chacun de ces marteaux fabrique, chaque jour, je ne sais combien de milliers de clous. Ce sont de jeunes filles fraîches et jolies qui présentent aux coups de ces marteaux énormes les petits morceaux de fer qui sont rapidement transformés en clous. Ce travail, si rude en apparence, est un de ceux qui étonnent le plus le voyageur qui pénètre pour la première fois dans les montagnes qui séparent la France de l'Helvétie. Si, en entrant à Verrières, le voyageur demande à qui appartient cette belle fabrique de clous qui assourdit les gens qui montent la grande rue, on lui répond avec un accent traînard : Eh ! elle est à M. le maire.
>
> Pour peu que le voyageur s'arrête quelques instants dans cette grande rue de Verrières, qui va en montant depuis la rive du Doubs jusque vers le sommet de la colline, il y cent à parier contre un qu'il verra paraître un grand homme à l'air affairé et important.

À son aspect tous les chapeaux se lèvent rapidement. Ses cheveux sont grisonnants, et il est vêtu de gris. Il est chevalier de plusieurs ordres, il a un grand front, un nez aquilin, et au total sa figure ne manque pas d'une certaine régularité : on trouve même, au premier aspect, qu'elle réunit à la dignité du maire de village cette sorte d'agrément qui peut encore se rencontrer avec quarante-huit ou cinquante ans. Mais bientôt le voyageur parisien est choqué d'un certain air de contentement de soi et de suffisance mêlé à je ne sais quoi de borné et de peu inventif. On sent enfin que le talent de cet homme-là se borne à se faire payer bien exactement ce qu'on lui doit, et à payer lui-même le plus tard possible quand il doit.

Tel est le maire de Verrières, M. de Rênal. Après avoir traversé la rue d'un pas grave, il entre à la mairie et disparaît aux yeux du voyageur. Mais, cent pas plus haut, si celui-ci continue sa promenade, il aperçoit une maison d'assez belle apparence, et, à travers une grille de fer attenante à la maison, des jardins magnifiques. Au-delà c'est une ligne d'horizon formée par les collines de la Bourgogne, et qui semble faite à souhait pour le plaisir des yeux. Cette vue fait oublier au voyageur l'atmosphère empestée des petits intérêts d'argent dont il commence à être asphyxié.

On lui apprend que cette maison appartient à M. de Rênal. C'est aux bénéfices qu'il a faits sur sa grande fabrique de clous que le maire de Verrières doit cette belle habitation en pierres de taille qu'il achève en ce moment. Sa famille, dit-on, est espagnole, antique, et, à ce qu'on prétend, établie dans le pays bien avant la conquête de Louis XIV.

Depuis 1815 il rougit d'être industriel : 1815 l'a fait maire de Verrières. Les murs en terrasse qui soutiennent les diverses parties de ce magnifique jardin qui, d'étage en étage, descend jusqu'au Doubs, sont aussi la récompense de la science de M. de Rênal dans le commerce du fer.

Ne vous attendez point à trouver en France ces jardins pittoresques qui entourent les villes manufacturières de l'Allemagne, Leipsick, Francfort, Nuremberg, etc. En Franche-Comté, plus on bâtit de murs, plus on hérisse sa propriété de pierres rangées les unes au-dessus des autres, plus on acquiert de droits aux respects de ses voisins. Les jardins de M. de Rênal, remplis de murs, sont encore admirés parce qu'il a acheté, au poids de l'or, certains petits morceaux de terrain qu'ils occupent. Par exemple, cette scie à bois, dont la position singulière sur la rive du Doubs vous a frappé en entrant à Verrières, et où vous avez remarqué le nom de Sorel, écrit en caractères gigantesques sur une planche qui domine le toit, elle occupait, il y a six ans, l'espace sur lequel on élève en ce moment le mur de la quatrième terrasse des jardins de M. de Rênal.

Malgré sa fierté, M. le maire a dû faire bien des démarches auprès du vieux Sorel, paysan dur et entêté ; il a dû lui compter de beaux louis d'or pour obtenir qu'il transportât son usine ailleurs. Quant au ruisseau public qui faisait aller la scie, M. de Rênal, au moyen du crédit dont il jouit à Paris, a obtenu qu'il fût détourné. Cette grâce lui vint après les élections de 182*.

Il a donné à Sorel quatre arpents pour un, à cinq cents pas plus bas sur les bords du Doubs. Et, quoique cette position fût beaucoup plus avantageuse pour son commerce de planches de sapin, le père Sorel, comme on l'appelle depuis qu'il est riche, a eu le secret d'obtenir de l'impatience et de la manie de propriétaire, qui animait son voisin, une somme de 6000 francs.

Le 53 avril – Jean-Philippe Dionne – Extrait

Planète Terre, an 2053. Le cœur du lieutenant Peter Schiller va finir par exploser tellement il bat fort. Plus fort que vite. Son moniteur n'indique en effet que 97 battements-minute. Malgré sa combinaison censée le tenir au frais, des gouttelettes dégoulinent le long de son épine dorsale, contre son siège qui le tient en étau. Impuissant, il suit le parcours lent et sinueux de chacune d'entre elles.

Il se tourne vers son supérieur, le commandant Thomas Major, qui s'active aux commandes du système de navigation. Le tableau de bord regorge de témoins qui clignotent et d'écrans où défile de l'information codée. Tout semble en ordre, tous les systèmes sont prêts. Un écran leur projette l'image de la rampe de lancement de l'Alveolo. Peter fige devant cette scène statique de leur vaisseau, ce que ne manque pas de remarquer le commandant Major :

- Ça va Peter ? Tu t'ennuies de ta maman ?
- Les vents soufflent à cinquante. On reporte à plus tard ?
- Thomas, se voulant rassurant, tapote l'ordinateur de bord :
- Tout baigne dans l'huile. Pas question d'annuler la mission !
- Je ne serai pas rassuré avant qu'on ait atteint l'espace.
- Patience ! Tu vas voir, on va être bien avec les autres dans le congélateur.
- Espérons que c'est bien resté " top secret ". Ça fait quand même cinq ans qu'on planifie le coup !
- Nos ennemis se battent plus entre eux que contre nous. C'est quoi les chances qu'ils aient réussi à repérer notre base et à tromper nos systèmes de détection ? Je doute qu'ils aient la moindre idée de l'existence même de notre projet.
- En cinq ans, ils auraient bien pu ! Puis c'est sans compter tous les jaloux qui auraient bien voulu eux aussi échapper à tout ce merdier.
- Ça je te l'accorde ! Plus du tout d'espoir sur Terre. D'ici vingt ans, plus un seul humain ici !

Un son long et aigu interrompt leur conversation. Le compte à rebours débute. Plus que trois minutes. Le radar ne signale toujours aucun mouvement au moins un kilomètre à la ronde. Les moteurs grondent.

- On est vraiment à la merci de terroristes, se plaint Peter.
- On ne pouvait pas se permettre de laisser du monde au courant de notre projet au sol, c'est tout.
- Les minutes les plus longues de ma vie ! En espérant que ce ne sont pas mes dernières !
- Relaxe Peter ! Tout est réglé au quart de tour. Vérifie donc pour moi les moteurs d'appoint. L'information défile trop vite ! Foutue idée de tout mettre en espéranto !
- Ne te plains pas, ç'aurait pu être du hongrois ! T'imagines !

Peter se plie à la requête du commandant et consulte nerveusement l'état du vaisseau. Pas d'anicroche. Un autre coup d'œil aux caméras de surveillance près de la structure de lancement. Toujours rien. Plus que trente secondes. Les deux hommes aux commandes demeurent vissés à leur siège. Thomas tente d'apaiser les tensions :

- Tu ne m'offrirais pas une bière ?

Trop absorbé par les grondements gagnant en intensité, Peter ignore la plaisanterie. Les vibrations envahissent leur minuscule habitacle. Plus que dix secondes. Il ferme les paupières et se concentre sur le battement de son cœur. Le décollage aura-t-il lieu sans sabotage ? L'équipement construit en quasi clandestinité tiendra-t-il le coup ?

Ça y est, l'Alveolo décolle ! Peter serre les dents, complètement écrasé sur place. Ses pensées se bousculent à une vitesse folle. Il songe à Anneke, qu'il sait étendue de tout son long dans son scaphandre congelé, comme les deux cents autres membres d'équipage. Ressent-elle tout ce brondissement ? Qu'est-ce qui mijote dans son cerveau au ralenti ? Il aurait tellement voulu partager ce moment avec elle.

Puis, penser devient trop pénible. Peter ne fait plus que subir. Le temps devient une abstraction. S'est-il écoulé une minute, dix minutes, ou même trente ?

Soudainement, le bruit s'apaise, les vibrations diminuent et la pression se relâche. Peter est tenté d'ouvrir les yeux. Une subtile odeur de brûlé finit de le réveiller. Une main saisit la sienne.

- On est parti ! Ça a marché !

Peter reprend ses esprits. Thomas, le visage tout rougi, exulte. Les caméras de surveillance au sol ne révèlent plus qu'un épais nuage gris informe. Peter sélectionne un autre point de vue. De l'extérieur du vaisseau, on devine déjà la courbure de la Terre. Le ciel passe du bleu au bleu foncé. Les deux pilotes sont rivés à l'écran. Thomas passe son premier commentaire :

- On ne la reverra plus jamais cette planète maudite !
- Surtout qu'on s'est juré de plus jamais prendre contact !
- De toute façon, ils vont tous s'entre-tuer.

Flottant dans le ciel noir étoilé, apparaît la Lune, presque pleine. Peter soupire :

- Dommage qu'elle n'ait pas pu nous accueillir, celle-là !
- Pas assez loin des barbares !
- Goya est mieux d'être comme on nous l'a décrite. Nous imaginerais-tu faire tout ça pour aboutir dans un demi-siècle sur une grosse roche sèche ?
- Détends-toi Peter, le pire est passé.

Un nouveau décompte s'enclenche : cinq minutes. Les deux hommes se taisent et inspectent les cadrans un à un. Un calme relatif s'installe dans leur capsule, maintenant que le module principal de l'Alveolo s'est détaché de l'énorme réservoir de combustible nucléaire. Peter est formel :

- J'ai tout vérifié. Rien de laissé à la chance. Tout fonctionne.
- Même chose de mon côté. C'est l'heure d'aller rejoindre nos amis.
- Vraiment dommage que Zalman ne soit pas de la partie. Quelle trahison !
- Avec trop de groupes ethniques, ça n'aurait pas pu marcher.
- Les embryons eux, t'en penses quoi ? s'enquiert Peter.
- Bof, on n'en aura sûrement pas besoin. On ne sait pas ce que ça donnerait une fois développé. Une autre idée excentrique du Dr. Zhang.

La Terre est maintenant visible en entier. Elle dérive en chute libre derrière eux. On y distingue l'étroite bande colorée de part et d'autre de l'équateur, coincée entre les deux énormes pôles de glace ayant submergé la majeure partie des continents. Peter se met à regretter d'avoir tout laissé derrière, peu rassuré par l'excès de confiance de son supérieur.

ANNEXE B – Caractéristiques grammaticales pour l'apprentissage machine

On énumère ici les 68 caractéristiques (« *features* ») utilisées pour l'apprentissage machine permettant la désambiguïsation d'homographes, dans le cas général. Ces caractéristiques, évaluées pour un mot en particulier, s'appuient sur la présence et la fonction des autres mots inclus au sein de la même phrase. L'utilisation de ces caractéristiques dans le contexte de l'apprentissage machine est décrite à la Section 4.6.

La plupart des caractéristiques décrites plus bas prennent ou bien la valeur de 1, si leur condition est vérifiée pour le mot sous étude, ou une valeur de zéro dans le cas contraire. Cet aspect binaire des caractéristiques n'est pas une condition pour l'application de l'apprentissage machine, mais reflète plutôt le type de caractéristique employé ici.

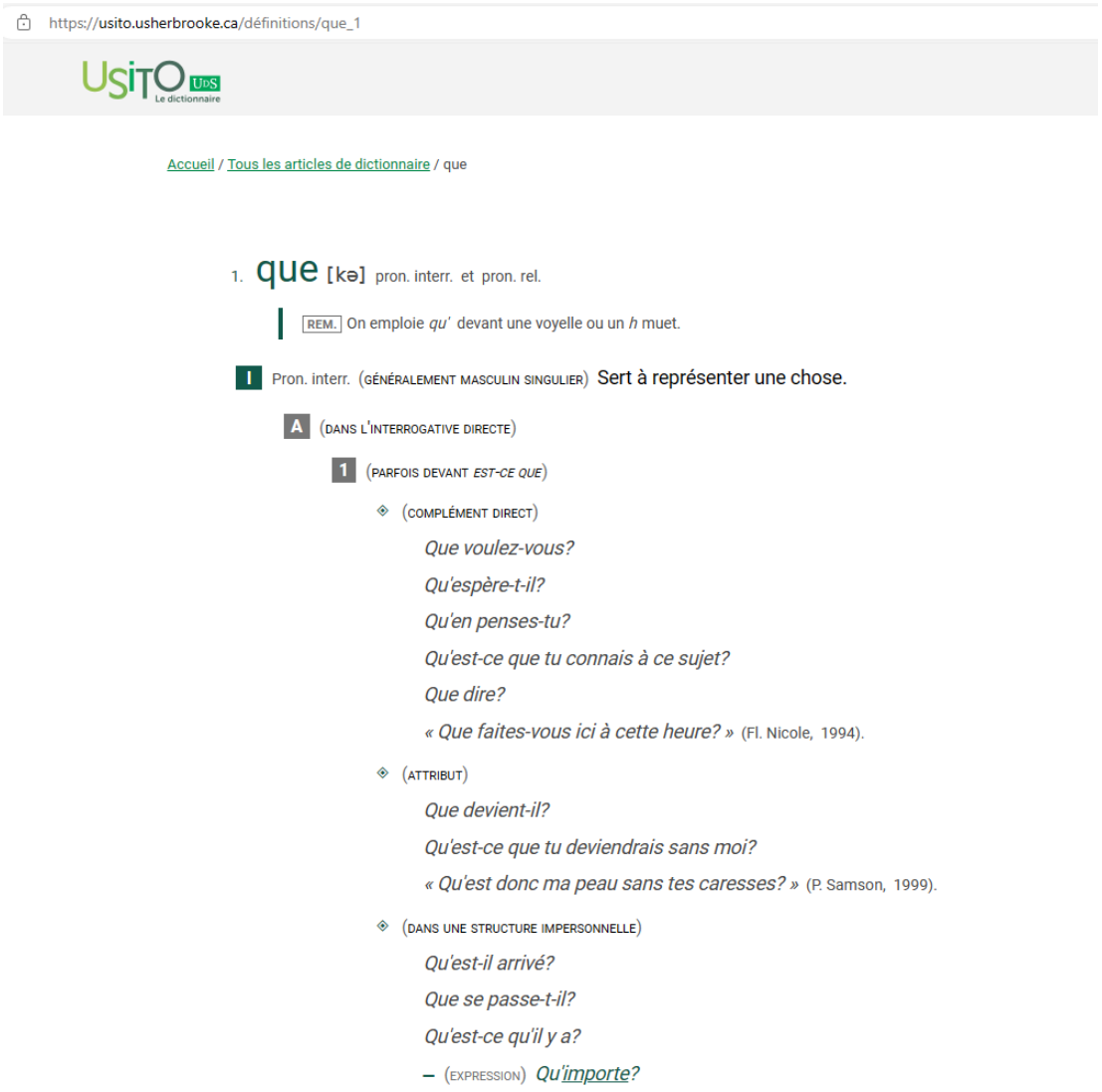
Numéro	Description de la caractéristique
1 à 9	Classe grammaticale du mot précédent, selon les codes de classes grammaticales fournis au Tableau 3.1. Par exemple, si le mot précédent est un verbe, la caractéristique 1 prend la valeur de 1 et les caractéristiques 2 à 9 prennent la valeur de zéro. Si le mot est le premier de la phrase, alors les caractéristiques 1 à 9 prennent toutes la valeur de zéro.
10 à 18	Classe grammaticale du mot situé deux mots en amont, selon les codes de classes grammaticales fournis au Tableau 3.1. Par exemple, si ce mot est un adjectif, la caractéristique 2 prend la valeur de 1 et les caractéristiques 1 et 3 à 9 prennent la valeur de zéro. Si le mot est le premier ou le second de la phrase, alors les caractéristiques 10 à 18 prennent toutes la valeur de zéro.
19 à 27	Classe grammaticale du mot suivant, selon les codes de classes grammaticales fournis au Tableau 3.1. Par exemple, si le mot suivant est un nom, la caractéristique 3 prend la valeur de 1 et les caractéristiques 1, 2 et 4 à 9 prennent la valeur de zéro. Si le mot est le dernier de la phrase, alors les caractéristiques 19 à 27 prennent toutes la valeur de zéro.
28 à 36	Classe grammaticale du mot situé deux mots en aval, selon les codes de classes grammaticales fournis au Tableau 3.1. Par exemple, si ce mot est un adverbe, la caractéristique 4 prend la valeur de 1 et les caractéristiques 1, 2, 3 et 5 à 9 prennent la valeur de zéro. Si le mot est le dernier ou l'avant-dernier de la phrase, alors les caractéristiques 28 à 36 prennent toutes la valeur de zéro.
37	Le mot sous étude est le premier de la phrase.
38	Le nombre de verbes conjugués inclus dans la phrase, excluant le mot sous étude. Pour les homographes non encore désambiguïsés, on considère la *possibilité* que chaque mot puisse être un verbe conjugué.
39	Le mot sous étude est peut-être un auxiliaire (forme du verbe « avoir » ou du verbe « être »), et on retrouve au moins un mot pouvant être un participe passé en aval dans la phrase.
40	Le mot sous étude est peut-être un verbe conjugué, et est situé entre les mots « ne » et « pas » (pas nécessairement directement avant et après) dans la phrase.
41	Le mot sous étude est peut-être un verbe à l'infinitif, et les mots « ne » et « pas » sont situés en amont dans la phrase (dans cet ordre).
42	Le mot sous étude est peut-être un nom, et on retrouve un mot pouvant être un adjectif dans les deux mots suivants. Cet adjectif doit s'accorder en genre et en nombre avec le nom.

43	Le mot sous étude est peut-être un nom, et on retrouve un mot pouvant être un adjectif antéposé directement devant. Cet adjectif doit s'accorder en genre et en nombre avec le nom.
44	Le mot sous étude est peut-être un nom, et on retrouve un mot pouvant être un adjectif antéposé directement devant. Cet adjectif ne s'accorde PAS en genre et en nombre avec le nom.
45	Le mot sous étude est peut-être un adjectif et est précédé par un mot pouvant être un verbe attributif, situé soit directement devant, ou séparé par un mot pouvant être un adverbe. Le nombre de l'adjectif doit concorder avec le nombre du verbe (pluriel ou singulier).
46	Le mot sous étude est peut-être un verbe à l'infinitif, et une des prépositions « à », « de », « pour » ou « sans » est située directement devant, ou séparée par un mot pouvant être un adverbe.
47	Le mot sous étude est peut-être un adjectif. Si l'adjectif peut être antéposé, on regarde si le mot *suivant* peut être un nom. Si l'adjectif n'est pas antéposé, on regarde si le mot *précédent* peut être un nom. On permet la présence d'un adverbe d'intensité dans le cas de l'adjectif antéposé. Le nom et l'adjectif ne concordent PAS en genre et en nombre.
48	Le mot sous étude peut être un déterminant, et il est suivi d'un mot pouvant être soit un adjectif numéral, un nom, ou un adjectif. Dans le cas de l'adjectif, permettre la présence d'un adverbe d'intensité. Le déterminant doit concorder en genre et en nombre avec le nom ou adjectif.
49	Le mot suivant le mot sous étude est un nom propre.
50	Le mot précédant le mot sous étude est un nom propre.
51	Le mot précédant le mot sous étude peut être un participe passé.
52	Le mot « en » précède le mot sous étude.
53	Le mot « en » suit le mot sous étude.
54	Le mot précédant le mot sous étude est un adjectif antéposé.
55	Le mot suivant le mot sous étude est un adjectif antéposé.
56	Le mot sous étude est directement précédé du mot « du », du mot « des » ou des mots « de la ».
57	On retrouve le mot « ne » n'importe où en aval du mot sous étude dans la phrase.
58	Le mot « et » précède le mot sous étude.
59	Le mot « et » suit le mot sous étude.
60	Le mot suivant le mot sous étude est un pronom démonstratif.
61	Le mot suivant le mot sous étude est un adverbe circonstanciel.
62	Le mot sous étude est directement précédé d'une virgule.
63	Le mot sous étude est directement suivi d'une virgule.
64	Le mot sous étude est directement suivi du mot « plus »
65	Le mot sous étude est directement suivi du mot « moins »
66	Le mot sous étude est directement précédé d'un trait d'union.
67	Le mot sous étude est directement suivi d'un trait d'union.
68	Le mot sous étude est directement précédé du mot « si »

ANNEXE C – Entrée du mot « que » dans le dictionnaire en ligne Usito

Le dictionnaire en ligne québécois « Usito » (2024) développé à l'Université de Sherbrooke, regorge d'exemples d'utilisation des mots qu'il définit. On illustre plus bas à titre d'exemple l'article du mot « que » utilisé comme pronom interrogatif et pronom relatif. Un autre article, non illustré ici, est en lien avec l'utilisation de ce mot comme conjonction de subordination.

De tels articles du dictionnaire « Usito » ont été abondamment utilisés dans ce mémoire pour extraire des caractéristiques en lien avec certains homographes. Ces caractéristiques ont ensuite été appliquées pour les « tests spécialisés » d'apprentissage machine (Section 4.6.5) pour faciliter la désambiguïsation de ces homographes.



◈ (POUR INDIQUER UNE QUANTITÉ)

Votre montre, que vaut-elle?

« Que pèse l'amour dans l'inapaisable sauvagerie de la vie? » (R. Lalonde, 1994).

2 (DEVANT *EST-CE QUI*)

◈ (SUJET) *Qu'est-ce qui brûle?*

B (DANS L'INTERROGATIVE INDIRECTE) (AVEC UN INFINITIF) *Nous ne savions plus que faire exactement.*

II Pron. rel. Sert à représenter une personne ou une chose.

1 (COMPLÉMENT DIRECT)

Les cadeaux qu'elle s'est offerts.

La promenade que vous avez faite.

◈ (NEUTRE) (L'ANTÉCÉDENT EST UNE PHRASE) *« Il vient de la montagne, et n'a pas de nom, que je sache »* (F.-A. Savard, 1959).

– FAM. *« Venez, qu'elle dit, venez vous déraidir un petit brin les jambes »* (Ant. Maillet, 1979).

◈ (DEVANT *VOICI*, *VOILÀ*)

La fille que voilà.

Le cadeau que voici est pour toi.

2 (ATTRIBUT)

L'adulte qu'elle deviendra.

Ils ne t'ont pas remarqué, occupés qu'ils sont.

3 (DANS UNE STRUCTURE IMPERSONNELLE) *La tempête qu'il y a eu.*

4 (COMPLÉMENT DE TEMPS OU DE MESURE)

Cela fait une éternité que je l'ai vu.

Les cinq kilomètres que j'ai couru.

5 (EXPRESSIONS)

Coûte que coûte.

Advienne que pourra.

VOIR l'article thématique TABLEAUX DES PRONOMS.

ÉTYMOLOGIE

842 (*in* TLF_i) pron. rel.; 2e moitié du 10e s. (*in* TLF_i) pron. interr.; du latin *quem*, accusatif de *qui*, pronom relatif, et de *quid*, accusatif neutre du pronom interrogatif *quis*.

ORTHOGRAPHE

	PRONOM INTERROGATIF	PRONOM RELATIF
que	**que** **(qu')**	**que** **(qu')**

ANNEXE D – Caractéristiques pour les tests spécialisés

À la Section 4.6.5, on mentionne que l'apprentissage machine a été appliqué directement à certains homographes dont la désambiguïsation sur la base des caractéristiques générales de l'Annexe B n'était pas efficace. Des caractéristiques (« *features* ») spécifiques à ces homographes ont été développées, favorisant une meilleure désambiguïsation pour ces cas particuliers. On présente ici la liste de toutes les caractéristiques utilisées pour ces quelques homographes ayant eu droit à ce traitement particulier.

Homographe : « que » ou « qu' »	
Numéro	Description de la caractéristique
1	Premier mot de la phrase
2	Premier mot après une virgule
3	Directement précédé du mot « ce »
4	Directement précédé d'un nom commun, ou nom et adjectif, ou nom et participe passé
5	Directement précédé du mot « advienne » ou du mot « coûte »
6	Directement précédé d'un verbe
7	Directement précédé d'une forme du verbe falloir
8	Directement précédé d'un adjectif ou d'un participe passé, mais eux-mêmes non précédés d'un nom commun
9	Directement précédé d'un adverbe
10	Utilisé dans les locutions : « avant que », « après que », « pendant que », « parce que », « attendu que », « autant que », « vu que », « afin que », « pour que », « pourvu que », « alors que », « sans que », « bien que », « moins que », « mieux que », « tel que », « et que », « dommage que », « espoir que », « de peur que », « de crainte que », « de sorte que », « à condition que », « le fait que », « à mesure que », « en sorte que », « une chance que », « grand-chance que », « sous prétexte que », « dans le sens que », « à la condition que »
11	Précédé dans la phrase (pas nécessairement directement) des mots : « même », « tant », « si », « tellement », « fait », « fois », « bien », « plus », « voilà » et « y »
12	Précédé dans la phrase (pas nécessairement directement) des mots « ne » ou « n' »
13	Précédé dans la phrase (pas nécessairement directement) de « c'est »
14	Directement précédé d'une préposition
15	Directement précédé d'un nom propre
16	Directement précédé des démonstratifs « celui », « celle », « ceux » ou « celles »
17	Suivi dans le reste de la phrase d'aucun verbe
18	Suivi dans les quelques mots suivants d'une forme verbale au subjonctif
19	Suivi dans les quelques mots suivants d'un auxiliaire avoir puis d'un participe passé qui s'accorde en genre et nombre avec sujet situé devant le « que »

Homographe : « de » ou « d’ »	
Numéro	Description de la caractéristique
1	Suivi par le mot « autres »
2	Suivi par un adjectif antéposé pluriel (potentiellement précédé d’un adverbe d’intensité), puis d'un nom au pluriel
3	Précédé d’un nom commun, avec option d’adverbe intensité et d’adjectif non antéposé qui s’accorde, mais non séparé par une virgule
4	Précédé d’un adverbe de quantification
5	Directement précédé d’une virgule
6	Premier mot de la phrase
7	Directement précédé du mot « né »
8	Directement suivi d’un nom propre
9	Directement suivi d’un pronom (sauf « autres »)
10	Directement suivi d’un adjectif numéral
11	Directement suivi d’un verbe à l’infinitif
12	Directement suivi d’un déterminant
13	Directement suivi d’un des mots suivants : « par », « tout », « ici », « avec », « plus », « est » et « ouest »
14	Précédé quelque part dans la phrase du mot « ne » mais le mot « que » non suivi d’un déterminant ou d’un infinitif
15	Même caractéristique que la précédente, mais excluant la présence du mot « ne »
16	Suivi d’un nom commun (optionnellement précédé d’un adverbe d’intensité et d’un adjectif antéposé qui s’accorde avec le nom), mais pas dans une phrase négative
17	Précédé d’un infinitif, mais pas suivi d’un infinitif
18	Directement suivi d’un verbe non à l’infinitif
19	Directement suivi d’une préposition

Homographe : « autre » ou « autres »	
Numéro	Description de la caractéristique
1	Mot « autres » directement précédé de « nous », « vous » ou « eux »
2	Directement suivi d’un nom commun (optionnellement précédé d’un adjectif antéposé) ou directement suivi d’un adjectif numéral
3	Directement suivi de « que », « qu’ » ou « qui »
4	Directement précédé du mot « entre »
5	Dernier mot de la phrase, ou directement suivi d’une virgule
6	Précédé du mot « en » dans les trois mots précédents

Homographe : « tout »	
Numéro	Description de la caractéristique
1	Directement suivi d’un déterminant et d’un nom commun (accompagné optionnellement d’un adverbe d’intensité et d’un adjectif antéposé)
2	Directement suivi d’un pronom démonstratif ou de « ce »
3	Directement précédé d’une virgule
4	Premier mot de la phrase
5	Directement suivi d'un nom commun s’accordant en genre et en nombre (accompagné optionnellement d’un adverbe intensité et d’un adjectif antéposé)
6	Directement suivi d’un nom propre
7	Directement précédé de « de », « en » ou « à », et directement suivi d’un nom commun s’accordant avec « tout »
8	Dans l’expression « tout un chacun »
9	Directement suivi d’un adjectif
10	Directement suivi d'un adverbe ou d'une préposition
11	Directement suivi du mot « en » puis d’un participe présent
12	Dans les expressions « tout de suite » et « tout de même »
13	Dans les expressions « tout à coup » et « tout à fait »
14	Dans la locution « de tout » en fin de phrase ou juste avant une virgule
15	Directement précédé d’un de ces mots : « est », « après », « malgré », « avant », « comme », « pas » ou « pour »
16	Directement suivi d’un verbe
17	Dernier mot de la phrase, ou juste avant une virgule
18	Directement précédé d’un déterminant

Homographe : « bien »	
Numéro	Description de la caractéristique
1	Directement suivi d’un adverbe
2	Premier mot de la phrase
3	Directement précédé d’un verbe
4	Directement précédé d’un de ces mots : « eh », « oh », « si », « et », « fort », ou « ou »
5	Directement suivi d’un déterminant
6	Directement suivi de « à » ou « des »
7	Suivi d'un adjectif ou d’un participe passé mais dont l’accord en genre et en nombre ne se fait pas avec le nom « bien »
8	Directement précédé d’un déterminant dont l’accord en genre et en nombre ne se fait pas avec le nom « bien »
9	Directement précédé d’un déterminant (sauf « de »), s’accorde avec le nom « bien »

Homographe : « fort »	
Numéro	Description de la caractéristique
1	Directement suivi d’un adverbe
2	Premier mot de la phrase
3	Directement précédé d’un verbe
4	Directement précédé du mot « si »
5	Directement suivi d’un déterminant
6	Directement suivi de « à » ou « comme »
7	Directement suivi d'un adjectif ou d’un participe passé mais qui ne s'accorde pas avec le nom « fort »
8	Directement précédé d’un déterminant, mais qui ne s’accorde pas avec le nom « fort »
9	Directement précédé d’un déterminant sauf « de », et s’accorde en genre et en nombre avec le nom « fort »

Homographe : « même »	
Numéro	Description de la caractéristique
1	Premier mot de la phrase
2	Directement précédé d’une virgule
3	Dernier mot de la phase, ou directement suivi d’une virgule
4	Directement suivi d’un déterminant
5	Directement suivi de « que », « qui », « de » ou « d’ »
6	Directement suivi d’une préposition
7	Directement suivi d’un adjectif, mais l’adjectif lui-même n’est pas directement suivi d’un nom
8	Directement précédé de « à », « de » ou « quand »
9	Directement suivi d’un nom (potentiellement accompagné d’un adjectif antéposé s’accordant)
10	Directement précédé du mot « en »
11	Directement précédé de « moi », « toi », « lui » ou « elle »
12	Directement précédé d’un nom commun qui s’accorde, mais pas séparé par une virgule
13	Directement précédé d’un déterminant qui s’accorde
14	Directement suivi d’un verbe
15	Directement précédé d’un verbe

Homographe : « si »	
Numéro	Description de la caractéristique
1	Directement suivi d’une virgule
2	Directement suivi d’un adverbe sauf ceux-ci : « jamais », « déjà » et « toutefois »
3	Directement suivi d’un adjectif
4	Directement suivi d’un participe passé
5	Premier mot de la phrase
6	Directement précédé d’une virgule
7	Directement suivi d’un déterminant
8	Directement suivi d’un pronom personnel nominatif (« je », « tu », etc.)
9	Directement précédé d’un de ces mots : « comme », « même », « sauf », ou « excepté »
10	Directement suivi d’un de ces mots : « en », « par », « jamais », « dans », « déjà » ou « toutefois »
11	Directement suivi d’un nom commun ou d’un nom propre

Homographe : « s’»	
Numéro	Description de la caractéristique
1	Directement suivi d’un verbe
2	Directement suivi de « en » ou « y »
3	Premier mot de la phase
4	Directement précédé d’une virgule
5	Directement suivi d'un pronom personnel nominatif (« je », « tu », etc.)
6	Directement précédé d’un de ces mots : « comme », « même », « sauf », ou « excepté »

ANNEXE E – Caractéristiques pour les participes passés

Pour faciliter la désambiguïsation des participes passés, qui peuvent tous, par définition, être aussi considérés comme des adjectifs (participes passés employés seuls), le paramètre *sommePP* est calculé. Si ce paramètre est supérieur à zéro, on élimine la possibilité que le mot sous étude puisse être un adjectif. Ce test en particulier ne s'effectue pas dans le contexte de l'apprentissage machine. Ce paramètre *sommePP* se calcule sur la base de 4 caractéristiques qui ne peuvent être égales qu'à un ou zéro. On additionne les trois premières, et on soustrait la dernière. Le paramètre *sommePP* peut donc avoir une valeur finale comprise entre -1 et 3, mais on ne vérifie que si sa valeur est supérieure à zéro. Les quatre caractéristiques utilisées sont listées plus bas.

Numéro	Description de la caractéristique
1	Le participe passé se conjugue avec l'auxiliaire « être » (selon le Bescherelle) et on retrouve effectivement un auxiliaire « être » devant le participe passé
2	Le participe passé est utilisé dans le contexte d'une forme pronominale (présence de « me », « te », « t' », « se » ou « s' » dans les mots précédents, mais un auxiliaire « être » doit aussi être présent
3	Présence de l'auxiliaire avoir devant le participe passé
4	Présence d'une forme de type « avoir été » suivi d'un participe passé (exemple : « avoir été mangé »)

ANNEXE F – Résultats détaillés des tests statistiques sur les verbes

À la Section 5.3.2.6, les résultats globaux pour l'efficacité des tests statistiques sur les verbes ont été fournis. Ces tests se sont avérés pertinents, puisqu'ils ont permis d'augmenter la performance globale de la désambiguïsation. Cependant, tel qu'illustré au Tableau 5.18, tandis que les tests ont permis de correctement classifier certains homographes qui étaient auparavant mal classifiés, ces tests ont aussi malheureusement, à quelques reprises, incorrectement classifié d'autres homographes qui étaient pourtant auparavant correctement classifiés. Quelques exemples de ces deux scénarios sont fournis plus bas.

Exemples de classifications correctes

Les tests statistiques servent à faire diminuer la probabilité qu'un homographe puisse être classifié comme un verbe, sur la base de certaines observations statistiques sur les verbes (fréquences de personnes et de temps). Tous les cas de classifications correctes impliquent donc des homographes qui ne sont *pas* des verbes, mais qui avaient été incorrectement classifiés comme verbes avant l'application des tests statistiques.

Le Tableau F1 fournit quelques exemples parmi les 48 homographes, auparavant incorrectement classifiés, qui ont été *correctement* classifiés après l'application des tests statistiques sur les verbes, au sein de la portion lemmatisée du roman « Le Rouge et le Noir ». Le tableau fournit l'homographe en jeu, suivi de la classe grammaticale maintenant correctement prédite, puis la valeur du paramètre « facteur verbe » décrit au Chapitre 4, suivi de la probabilité que ce mot soit un verbe, calculée par l'algorithme en appliquant le test statistique. Toutes les valeurs de cette colonne sont inférieures à 50%, puisque tous les homographes du Tableau F1 appartiennent à des classes grammaticales autres que « verbe ». La colonne suivante (en jaune), illustre cette même probabilité, obtenue avant d'appliquer le test statistique sur les verbes. Toutes les valeurs de cette colonne sont supérieures à 50%, puisque tous ces homographes avaient auparavant été incorrectement classifiés comme étant des verbes. Les deux probabilités de verbe et le « facteur verbe » sont reliés par l'équation suivante :

$$Prob_{avec\ tests} = Prob_{sans\ tests} \times Facteur_{verbe}$$

En appliquant cette équation à la première ligne du Tableau F1, on obtient :

$$31\% = 55\% \times 0.56$$

Les phrases complètes où se retrouvent ces homographes sont fournies à la dernière colonne. L'homographe en jeu est indiqué en lettres majuscules pour permettre de facilement le repérer.

On remarque au Tableau F1 que plusieurs des probabilités de verbe avant l'application des tests statistiques (colonnes jaunes) n'étaient que de peu supérieures à 50%. On en déduit que l'algorithme général de désambiguïsation en lui-même n'était donc pas trop loin d'une bonne prédiction. On constate aussi que l'algorithme a aidé à classifier deux mauvaises prédictions de participes présent (« maintenant » et « sanglant »). Dans le cas de l'homographe « costume », la présence du mot « en » tout juste devant a pu influencer la prédiction initiale erronée.

Bien que non mentionné au Tableau F1, il est à noter que la classification correcte du deuxième homographe « plus » a mené à une classification correcte de l'homographe suivant « haut ». On parle donc ici d'effet de levier, car une correction directe a mené à une correction indirecte supplémentaire.

Tableau F1 : Exemples d'homographes auparavant incorrectement classifiés, qui ont été correctement classifiés en appliquant le test statistique sur les verbes

		Avec test statistique pour les verbes		Sans test statistique	
Mot	Classe réelle	Facteur Verbe	Probabilité Verbe	Probabilité Verbe	Phrase complète
terrasse	Nom	0.56	31%	55%	Les murs en TERRASSE qui soutiennent les diverses parties de ce magnifique jardin qui, d'étage en étage, descend jusqu'au Doubs, sont aussi la récompense de la science de monsieur de Rênal dans le commerce du fer.
plus	Adverbe	0.68	34%	50%	En Franche-Comté, plus on bâtit de murs, plus on hérisse sa propriété de pierres rangées les unes au-dessus des autres, PLUS on acquiert de droits aux respects de ses voisins.
costume	Nom	0.56	44%	78%	Une fois, c'était un jour de dimanche, il y a quatre ans de cela, monsieur de Rênal, revenant de l'église en COSTUME de maire, vit de loin le vieux Sorel, entouré de ses trois fils, sourire en le regardant.
maintenant	Adverbe	0.38	24%	65%	Le parapet de ce mur pour lequel monsieur de Rênal a dû faire trois voyages à Paris, car l'avant-dernier ministre de l'Intérieur s'était déclaré l'ennemi mortel de la promenade de Verrières, le parapet de ce mur s'élève MAINTENANT de quatre pieds au-dessus du sol.
cascade	Nom	0.56	33%	58%	Après avoir couru de cascade en CASCADE on les voit tomber dans le Doubs.
bois	Nom	0.75	41%	55%	Le toit est soutenu par une charpente qui porte sur quatre gros piliers en BOIS.
plus	Adverbe	0.68	39%	57%	Il l'aperçut à cinq ou six pieds PLUS haut, à cheval sur l'une des pièces de la toiture.
violent	Adjectif	0.28	15%	53%	Un coup violent fit voler dans le ruisseau le livre que tenait Julien, un second coup aussi VIOLENT, donné sur la tête, en forme de calotte, lui fit perdre l'équilibre.
sanglant	Adjectif	0.28	19%	66%	Julien, quoique étourdi par la force du coup, et tout SANGLANT, se rapprocha de son poste officiel, à côté de la scie.
solde	Nom	0.56	31%	55%	En mourant, il lui avait légué sa croix de la Légion d'honneur, les arrérages de sa demi-SOLDE et trente ou quarante volumes, dont le plus précieux venait de faire le saut dans le ruisseau public, détourné par le crédit de monsieur le maire.

Exemples de classifications incorrectes

Tel que précisé plus haut, les tests statistiques ne servent qu'à faire diminuer la probabilité qu'un homographe puisse être classifié comme un verbe. Tous les cas de classifications incorrectes impliquent donc des homographes qui sont des verbes, et qui avaient au départ été correctement classifiés comme tel, mais dont la classification s'est malheureusement retrouvée erronée suite à l'application des tests statistiques.

Le Tableau F2 liste les 11 homographes, auparavant correctement classifiés, qui ont été incorrectement classifiés après l'application du test statistique sur les verbes, au sein de la portion lemmatisée du roman « Le Rouge et le Noir ». Le tableau fournit l'homographe en jeu, suivi de la valeur du paramètre « facteur verbe » décrit au Chapitre 4, suivi de la probabilité que ce mot soit un verbe, calculée par l'algorithme en appliquant les tests statistiques. La colonne suivante (en jaune), illustre cette même probabilité, obtenue avant d'appliquer les tests statistiques sur les verbes. Le lien mathématique entre les deux probabilités et le « facteur verbe » est le même que décrit plus haut au sujet du Tableau F1. Les phrases complètes où se retrouvent ces homographes sont fournies à la dernière colonne. L'homographe en jeu est indiqué en lettres majuscules pour permettre de facilement le repérer.

On constate que dans les deux cas impliquant l'homographe « dit », la probabilité de forme verbale avant d'appliquer les tests statistiques sur les verbes était déjà tout juste au-dessus de 50%, ce qui veut dire qu'il ne s'agissait pas de prédictions « fortes ». Un « rien » aurait pu les faire basculer, comme ce fut malheureusement le cas ici.

Dans le cas de l'homographe « sourire », le verbe est séparé de son sujet (« le vieux Sorel ») par une proposition entourée de virgules, ce qui rend l'analyse plus complexe. Dans le cas de l'homographe « triplé », il s'agit d'un mot très peu utilisé dans le corpus, si bien que les statistiques n'ont pas pu être basées sur un nombre suffisant d'observations. Considérant l'homographe « employés », on constate au moins que l'algorithme de base, avant l'application du test statistique, offrait une haute probabilité pour la forme verbale (prédiction correcte). Aussi, on constate que 6 des 11 cas du Tableau F2 concernent une forme verbale à l'infinitif. Le test statistique aurait peut-être pu être ajusté pour tenir compte de ces formes en particulier.

On peut finalement se consoler du fait que les 11 erreurs de classifications directement causées par l'application des tests statistiques n'ont pas engendré d'erreurs supplémentaires de classification pour les homographes environnants, ce qui aurait bien pu se produire, considérant que les caractéristiques de chaque homographe utilisées en apprentissage machine dépendent fortement des mots environnants, en particulier de leur classe grammaticale.

Tableau F2 : Tous les cas d'homographes auparavant correctement classifiés, qui ont été incorrectement classifiés en appliquant le test statistique sur les verbes

	Avec test statistique pour les verbes		Sans test statistique	
Mot	Facteur Verbe	Probabilité Verbe	Probabilité Verbe	Phrase complète
sourire	0.56	33%	59%	Une fois, c'était un jour de dimanche, il y a quatre ans de cela, monsieur de Rênal, revenant de l'église en costume de maire, vit de loin le vieux Sorel, entouré de ses trois fils, SOURIRE en le regardant.
repentir	0.56	41%	72%	Il pourrait bien s'en REPENTIR, ce beau monsieur de Paris, disait monsieur de Rênal d'un air offensé, et la joue plus pâle encore qu'à l'ordinaire
dit	0.95	48%	51%	Qui a pu mettre ce papier-là, DIT Julien.
triplé	0.56	33%	59%	S'écria-t-il, on dirait que c'est la seule vertu, et cependant quelle considération, quel respect bas pour un homme qui évidemment a doublé et TRIPLÉ sa fortune, depuis qu'il administre le bien des pauvres.
dit	0.95	48%	51%	Voilà une idée qui n'est pas mal, DIT monsieur de Rênal, évidemment fort joyeux.
employés	0.28	23%	82%	Il trouvait les mots qu'eût EMPLOYÉS un jeune séminariste fervent, mais le ton dont il les prononçait, mais le feu mal caché qui éclatait dans ses yeux alarmaient monsieur Chélan.
dîner	0.56	43%	76%	Jamais vous n'avez été si jeune, madame, lui disaient ses amis de Verrières qui venaient DÎNER à Vergy.
sourire	0.56	37%	65%	Cette position physique le fit SOURIRE, elle lui peignait la position qu'il brûlait d'atteindre au moral.
supporter	0.56	33%	58%	Ce surcroît de douleur arriva à toute l'intensité de malheur qu'il est donné à l'âme humaine de pouvoir SUPPORTER.
souffrant	0.28	24%	84%	Enfin, SOUFFRANT plus mille fois que s'il eût marché à la mort, il entra dans le petit corridor qui menait à la chambre de madame de Rênal.
dîner	0.56	39%	69%	Julien comprit enfin les demi-mots qu'il avait surpris, quand la haute société du pays venait DÎNER chez monsieur de Rênal.

ANNEXE G – Résultats plus détaillés de l'algorithme de phrase complète

À la Section 5.3.2.8, les résultats globaux pour l'efficacité de l'approche de désambiguïsation basée sur la phrase complète ont été fournis. Ce test ne s'est *pas* avéré pertinent, puisqu'il a fait *diminuer* la performance globale de la désambiguïsation, en comparaison avec l'approche dite « gauche-droite » (décrite à la Section 4.6.4.3). En effet, tel qu'illustré au Tableau 5.20 pour le roman « Le Rouge et le Noir », bien que le test ait permis de correctement classifier certains homographes qui étaient auparavant mal classifiés (95 cas), ce test a malheureusement, à davantage de reprises, incorrectement classifié d'autres homographes qui étaient pourtant auparavant correctement classifiés (276 cas). Quelques exemples de ces deux scénarios sont fournis plus bas.

Exemples de classifications correctes

Au Tableau G1, on liste certains homographes issus de la portion lemmatisés du roman « Le Rouge et le Noir », parmi les 95 cas disponibles qui ont été correctement classifiés grâce à l'analyse basée sur la phrase complète, alors qu'ils étaient auparavant incorrectement classifiés selon l'approche dite « gauche-droite ». Le tableau fournit l'homographe en jeu, suivi de la classe grammaticale maintenant correctement prédite, ainsi que la probabilité calculée en lien avec cette prédiction (cases vertes). Toutes ces valeurs sont supérieures à 0.50 (50%), ce qui explique que la classe en question ait été assignée. Pour chaque homographe, le tableau fournit ensuite (cases jaunes) la classe grammaticale prédite selon l'approche gauche-droite, accompagnée de la probabilité associée. Toutes ces valeurs sont aussi supérieures à 0.50, sauf une. En effet, pour l'homographe « droit », la probabilité n'est que 0.37. C'est que pour cet homographe, il y a trois possibilités : adjectif, nom et adverbe. Il appert que la valeur de 0.37 est la plus élevée des trois. Dans tous les cas du Tableau G1, la classe grammaticale appropriée pour le texte correspond à celle prédite en considérant la phrase complète.

La première phrase implique deux homographes qui se suivent : « favoris noirs ». On fait face ici à un cas assez fréquent où deux homographes pouvant être un adjectif ou un nom se suivent. On pourrait interpréter ces deux mots comme représentant « un favori qui est noir », ou encore « un noir qui est favori ». L'algorithme basé sur la phrase complète a su correctement distinguer le cas le plus probable, ce qui n'a pas été le cas de l'algorithme gauche-droite. La phrase suivante implique quatre homographes se suivant directement dans la même phrase. Avec l'algorithme gauche-droite, ces quatre homographes ont été mal classifiés, dans un effet de domino. C'est que cet algorithme ne peut assigner de valeur unique et précise à tout homographe en aval, si bien que les prédictions s'en retrouvent affectées. Cette phrase est un exemple parfait de ce à quoi on s'attendait exactement de l'algorithme avec phrase complète, qui a correctement réussi à prédire ces quatre homographes, en testant toutes les combinaisons possibles.

L'homographe « remplis », à l'exemple suivant, est suivi des mots « de larmes », qui peuvent être un déterminant suivi d'un nom. Comme un verbe précède souvent un déterminant suivi d'un nom (« je *mange une pomme* »), l'algorithme gauche-droite a incorrectement associé l'homographe « remplis » à un verbe (il s'agit plutôt d'un participe passé employé seul, associé pour ce projet à un adjectif). Mais encore une fois ici, l'algorithme basé sur la phrase complète, en considérant toutes les possibilités, a su bien classifier ces homographes.

Le cas de l'homographe « droit » est plus subtil. Tout lecteur peut deviner qu'il s'agit ici d'un adjectif (bras droit), mais pour l'algorithme, qui ne si connait pas du tout en sémantique, l'homographe « droit » peut aussi bien être un adverbe, qui voudrait dire que le bras se tient

« directement » contre la poitrine. Toujours est-il que l'algorithme basé sur la phrase complète a su bien classifier l'homographe « droit ».

La dernière phrase implique deux homographes qui se suivent et qui peuvent être combinés de quatre façons différentes. Dans la phrase en question, « le sale » consiste en un déterminant suivi d'un adjectif. Mais l'algorithme gauche-droite a plutôt interprété ces mots comme étant le pronom « le » suivi d'une forme conjuguée du verbe « saler ». Encore une fois, l'algorithme de phrase complète a su bien classifier ces homographes.

On retrouve donc au Tableau G1 quelques exemples typiques de ce qu'on recherchait avec l'algorithme de phrase complète, soit la capacité de mieux considérer les homographes en aval (plus loin dans la phrase), ce que ne permet pas l'approche gauche-droite.

Tableau G1 : Exemples d'homographes auparavant incorrectement classifiés, qui ont été correctement classifiés en appliquant l'approche basée sur la phrase complète

	Approche phrase complète		Approche gauche-droite		
Mot	Classe prédite (correcte)	Prob.	Classe prédite (incorrecte)	Prob.	Phrase complète
favoris	Nom	1.0	Adjectif	0.51	Monsieur Valenod, le riche directeur du dépôt, passait pour lui avoir fait la cour, mais sans succès, ce qui avait jeté un éclat singulier sur sa vertu, car ce monsieur Valenod, grand jeune homme, taillé en force, avec un visage coloré et de gros FAVORIS NOIRS, était un de ces êtres grossiers, effrontés et bruyants, qu'en province on appelle de beaux hommes.
noirs	Adjectif	0.59	Nom	0.87	
le	Pronom	0.99	Déterminant	0.79	Elle supposait sans se LE DIRE QU'ENTRE mari et femme il n'y avait pas de plus douces relations.
dire	Verbe	1.0	Nom	0.82	
qu'	Conjonction	0.89	Pronom	0.74	
entre	Préposition	1.0	Verbe	0.56	
remplis	Adjectif	0.56	Verbe	0.66	Les grands yeux noirs et REMPLIS de larmes de Julien se trouvèrent en face des petits yeux gris et méchants du vieux charpentier, qui avait l'air de vouloir lire jusqu'au fond de son âme.
droit	Adjectif	0.73	Adverbe	0.37	Il se lia le bras DROIT contre la poitrine, prétendit s'être disloqué le bras en remuant un tronc de sapin, et le porta pendant deux mois dans cette position gênante.
le	Déterminant	0.65	Pronom	0.57	Je pensais, monsieur, lui dit-il un jour, qu'il y aurait une haute inconvenance à ce que le nom d'un bon gentilhomme tel qu'un Rênal parût sur LE SALE registre du libraire.
sale	Adjectif	1.0	Verbe	0.64	

Exemples de classifications incorrectes

Au Tableau G2, on liste certains homographes issus de la portion lemmatisée du roman « Le Rouge et le Noir », parmi les 276 cas disponibles qui ont été *incorrectement* classifiés en se basant sur la phrase complète, alors qu'ils étaient auparavant correctement classifiés selon l'approche dite « gauche-droite ». Le tableau fournit l'homographe en jeu, suivi de la classe grammaticale maintenant incorrectement prédite, ainsi que la probabilité calculée en lien avec cette prédiction (cases vertes). Toutes ces valeurs sont supérieures à 0.50 (50%), ce qui explique que la classe en question ait été assignée. Pour chaque homographe, le tableau fournit ensuite (cases jaunes) la classe grammaticale prédite selon l'approche gauche-droite, accompagnée de la probabilité associée. Toutes ces valeurs sont aussi supérieures (ou égales) à 0.50. Dans tous les cas du Tableau G2, la classe grammaticale appropriée pour le texte correspond à celle prédite en considérant l'approche « gauche-droite ».

Le Tableau G2 compte quatre exemples de paires d'homographes qui se suivent directement, qui étaient correctement classifiés avec la méthode gauche-droite, mais qui se sont retrouvés à être mal classifiés avec l'algorithme basé sur la phrase complète (« majeure partie », « fort chaud », « la place », et « le livre »). Ces cas impliquent différentes combinaisons de classes grammaticales (adjectif-nom, adverbe-adjectif et déterminant-nom). Le fait que l'algorithme gauche-droite a mieux fonctionné pour ces cas précis tient peut-être du fait que de façon générale, les locuteurs bâtissent leur phrase de gauche à droite, tout comme cet algorithme.

Dans le cas de l'homographe « son », on voit que celui-ci est placé directement devant un adjectif (« immense »), lui-même suivi d'un nom (« mur »), ce qui est typique des mots suivants un déterminant. Toujours est-il que la prédiction correcte pour la méthode gauche-droite dans ce cas-ci a été obtenue avec une probabilité limite de tout juste 0.5. La méthode avec phrase complète a fait basculer la probabilité dans l'autre sens, mais demeure faible à 0.57.

Dans le cas de l'homographe « pas », l'algorithme avec phrase complète a dû avoir été influencé par l'homographe « prouvé » situé tout juste après lui, ici un participe passé employé seul (donc utilisé comme un adjectif). Un nom précède souvent un adjectif.

Ces quelques exemples démontrent que l'algorithme basé sur la phrase complète nécessiterait probablement l'ajout d'autres caractéristiques permettant de correctement classifier les homographes. En effet, les caractéristiques choisies pour ce projet (telles que décrites à l'Annexe B), l'ont été dans le but de maximiser la performance de l'approche « gauche-droite ». Un travail d'optimisation serait requis pour améliorer la performance de l'approche basée sur la phrase complète, mais cet effort va au-delà de la portée du présent projet, surtout considérant la performance tout à fait convenable de l'approche gauche-droite (de l'ordre de 95%).

Tableau G2 : Exemples d'homographes auparavant correctement classifiés, qui ont été incorrectement classifiés en appliquant l'approche basée sur la phrase complète

	Approche phrase complète		Approche gauche-droite		
Mot	Classe prédite (incorrecte)	Prob.	Classe prédite (correcte)	Prob.	Phrase complète
majeure	Nom	0.94	Adjectif	0.55	Un torrent, qui se précipite de la montagne, traverse Verrières avant de se jeter dans le Doubs, et donne le mouvement à un grand nombre de scies à bois, c'est une industrie fort simple et qui procure un certain bien-être à la MAJEURE PARTIE des habitants plus paysans que bourgeois.
partie	Adjectif	0.78	Nom	0.64	
été	Nom	0.74	Verbe	0.87	Il est vrai que cet arrangement a ÉTÉ critiqué par les bonnes têtes de l'endroit.
fort	Adjectif	1.0	Adverbe	1.0	Le soleil est FORT CHAUD dans ces montagnes, lorsqu'il brille d'aplomb, la rêverie du voyageur est abritée sur cette terrasse par de magnifiques platanes.
chaud	Nom	0.91	Adjectif	0.99	
son	Nom	0.57	Déterminant	0.50	Leur croissance rapide et leur belle verdure tirant sur le bleu, ils la doivent à la terre rapportée, que monsieur le maire a fait placer derrière SON immense mur de soutènement, car, malgré l'opposition du conseil municipal, il a élargi la promenade de plus de six pieds quoiqu'il soit ultra et moi libéral, je l'en loue, c'est pourquoi dans son opinion et dans celle de monsieur Valenod, l'heureux directeur du dépôt de mendicité de Verrières, cette terrasse peut soutenir la comparaison avec celle de Saint-Germain-en-Laye.
pas	Nom	0.90	Adverbe	1.0	Cet homme pouvait fort bien n'être au fond qu'un agent secret des libéraux, il disait que l'air de nos montagnes faisait du bien à son asthme, mais c'est ce qui n'est PAS prouvé.
la	Pronom	0.96	Déterminant	0.82	Celui-ci se dirigea vers le hangar, en y entrant, il chercha vainement Julien à LA PLACE qu'il aurait dû occuper, à côté de la scie.
place	Verbe	0.82	Nom	0.92	
le	Pronom	1.0	Déterminant	0.74	Un coup violent fit voler dans le ruisseau LE LIVRE que tenait Julien, un second coup aussi violent, donné sur la tête, en forme de calotte, lui fit perdre l'équilibre.
livre	Verbe	0.90	Nom	0.89	

ANNEXE H – Exemples d'erreurs de désambiguïsation

A la Section 5.3.4, on discute des sources d'erreurs de désambiguïsation les plus fréquentes observées lors de l'analyse des deux corpus de référence. Celles-ci ont été plus ou moins arbitrairement regroupées au Tableau 5.20 en 10 catégories distinctes. Dans ce tableau, on fournit les proportions d'erreurs (sous forme de pourcentage) associées à chaque catégorie. Dans cette annexe, on fournit quelques exemples pour chacune de ces sources d'erreur. Pour plus de détails, se référer à la Section 5.3.4.

Dans les exemples suivants, l'homographe incorrectement désambiguïsé est écrit en lettres majuscules, pour bien le distinguer. On précise aussi, pour chaque exemple choisi, la classe grammaticale incorrectement prédite, suivie de la classe réelle dans le contexte du corpus.

Cas limite (presque réussi)

Phrase : J'aime l'ombre, répondit monsieur de Rênal avec la nuance de hauteur convenable quand on parle à un chirurgien, membre de la Légion d'honneur, j'aime l'ombre, je fais tailler mes arbres pour donner de l'ombre, et je ne conçois pas qu'un arbre soit fait pour autre chose, quand toutefois, comme **L**'utile noyer, il ne rapporte pas de revenu.

Homographe : « l' »

Classe prédite : pronom (probabilité 0.51)

Classe réelle : déterminant

Commentaire : La présence de l'homographe « noyer » (nom ou verbe) n'a sans doute pas aidé la chose.

Phrase : Monsieur Valenod, le riche directeur du dépôt, passait pour lui avoir fait la cour, mais sans succès, ce qui avait jeté un éclat singulier sur sa vertu, car ce monsieur Valenod, grand jeune homme, taillé en force, avec un visage coloré et de gros **FAVORIS** noirs, était un de ces êtres grossiers, effrontés et bruyants, qu'en province on appelle de beaux hommes.

Homographe : « favoris »

Classe prédite : adjectif (probabilité 0.51)

Classe réelle : nom

Commentaire : Le fait que de façon générale les homographes « favoris » et « noirs » puissent tous les deux être des adjectifs n'a pas aidé.

Phrase : Mais à peine hors de la vue de **SON** terrible père, il ralentit le pas.

Homographe : « son »

Classe prédite : nom (probabilité 0.50)

Classe réelle : déterminant

Commentaire : L'adjectif « terrible » aurait bien pu en effet se rapporter au nom commun « son ».

Structure de phrase inhabituelle

Phrase : **QU**'importe!

Homographe : « qu' »

Classe prédite : Pronom

Classe réelle : Conjonction

Commentaire : Phrase très courte, dont le « sujet » est très court « qu' » et sans complément.

Phrase : Reprit vivement le geôlier, vous, monsieur le curé, on sait que vous avez **LIVRES** de rente, du bon bien au soleil.

Homographe : « livres »

Classe prédite : Verbe

Classe réelle : Nom

Commentaire : Il est inhabituel pour un nom commun de ne pas être précédé d'un déterminant.

Phrase : Je lui donnerai **FRANCS** et la nourriture.

Homographe : « francs »

Classe prédite : Adjectif

Classe réelle : Nom

Commentaire : Il est inhabituel pour un nom commun de ne pas être précédé d'un déterminant.

Phrase complexe

Phrase: A peine entre-t-on dans la ville **QUE** l'on est étourdi par le fracas d'une machine bruyante et terrible en apparence.

Homographe : « que »

Classe prédite : Pronom

Classe réelle : Conjonction

Commentaire : L'homographe « que » directement précédé d'un nom commun est souvent un pronom. L'analyse de la phrase ici requiert une compréhension sémantique.

Phrase: Malgré sa fierté, monsieur le maire a dû faire bien des démarches auprès du vieux Sorel, **PAYSAN** dur et entêté, il a dû lui compter de beaux louis d'or pour obtenir qu'il transportât son usine ailleurs.

Homographe : « paysan »

Classe prédite : Adjectif

Classe réelle : Nom

Commentaire : Le nom « paysan » n'est pas précédé d'un déterminant. De plus, il n'est pas précédé d'un verbe attributif, et n'est pas non plus l'objet d'un verbe.

Phrase: Leur croissance rapide et leur belle verdure tirant sur le bleu, ils la doivent à la terre rapportée, que monsieur le maire a fait placer derrière son immense mur de soutènement, car, malgré l'opposition du conseil municipal, il a élargi la promenade de plus de six pieds quoiqu'il soit ultra et moi **LIBÉRAL**, je l'en loue, c'est pourquoi dans son opinion et dans celle de monsieur Valenod, l'heureux directeur du dépôt de mendicité de Verrières, cette terrasse peut soutenir la comparaison avec celle de Saint-Germain-en-Laye.

Homographe : « libéral »

Classe prédite : Adjectif

Classe réelle : Nom

Commentaire : Le verbe être n'est pas répété après la conjonction « et ». De plus, il n'est jamais clair si la conjonction « et » introduit une nouvelle proposition ou plutôt une énumération.

Homographes environnants mal désambiguïsés

Phrase: Elle avait un certain air de simplicité, et de la jeunesse dans la démarche, aux yeux d'un parisien, cette grâce naïve, pleine d'innocence et de vivacité, serait **même ALLÉE** jusqu'à rappeler des idées de douce volupté.

Homographe : « allée »

Classe prédite : Nom

Classe réelle : Verbe (participe passé)

Commentaire : L'homographe « même » situé tout juste devant avait été incorrectement classifié comme un adjectif (plutôt qu'un adverbe), favorisant la classification de « allée » comme un nom.

Phrase: Elle supposait sans se **le DIRE** qu'entre mari et femme il n'y avait pas de plus douces relations.

Homographe : « dire »

Classe prédite : Nom

Classe réelle : Verbe

Commentaire : L'homographe « le » situé tout juste devant avait été incorrectement classifié comme un déterminant (plutôt qu'un pronom), favorisant la classification de « dire » comme un nom.

Phrase: Puisque Sorel n'est pas ravi et comblé de ma proposition, comme naturellement il devrait **I'ÊTRE**, il est clair, se dit-il, qu'on lui a fait des offres d'un autre côté, et de qui peuvent-elles venir, si ce n'est du Valenod.

Homographe : « être »

Classe prédite : Nom

Classe réelle : Verbe

Commentaire : L'homographe « l' » situé tout juste devant avait été incorrectement classifié comme un déterminant (plutôt qu'un pronom), favorisant la classification de « être » comme un nom.

Pourrait être bien classifié avec ajout d'une locution

Phrase: Se tournant **TOUT de suite** vers le monsieur de paris, avec des yeux où, malgré le grand âge, brillait ce feu sacré qui annonce le plaisir de faire une belle action un peu dangereuse.

Homographe : « tout »

Classe prédite : Déterminant

Classe réelle : Adverbe

Commentaire : La locution « tout de suite » a été considérée uniquement comme une « caractéristique », au lieu de comme une locution à toujours respecter. Les caractéristiques influencent le résultat, mais ne le déterminent pas.

Phrase: Monsieur appert comprit qu'il **avait AFFAIRE** à un homme de cœur.

Homographe : « affaire »

Classe prédite : Verbe

Classe réelle : Nom

Commentaire : La locution « avoir affaire » pourrait être intégrée à la liste de locutions. Il faudra considérer le lemme « avoir » situé devant le nom « affaire », puisque plusieurs formes verbales du verbe « avoir » sont possibles pour cette locution

Phrase : Monsieur, **DIT-il** au curé, dès qu'il l'aperçut, ce monsieur que je vois là avec vous, n'est-il pas monsieur Appert.

Homographe : « dit »

Classe prédite : Adjectif

Classe réelle : Verbe

Commentaire : La suite de mots « dit-il » est si courante dans un roman contenant des dialogues, qu'il vaudrait la peine de la considérer comme une « locution » déterministe.

Adjectif antéposé interprété comme un nom commun

Phrase : Malgré sa fierté, monsieur le maire a dû faire bien des démarches auprès du **VIEUX** Sorel, paysan dur et entêté, il a dû lui compter de beaux louis d'or pour obtenir qu'il transportât son usine ailleurs.

Homographe : « vieux »

Classe prédite : Nom

Classe réelle : Adjectif (antéposé)

Commentaire : On observe plusieurs cas semblables dans le roman. Il faudrait incorporer une ou des caractéristiques dédiées à ce problème de classification.

Phrase: Attentif à copier les habitudes des gens de cour, dès les **PREMIERS** beaux jours du printemps, monsieur de Rênal s'établit à Vergy, c'est le village rendu célèbre par l'aventure tragique de Gabrielle.

Homographe : « premiers »

Classe prédite : Nom

Classe réelle : Adjectif (antéposé)

Commentaire : On observe plusieurs cas semblables dans le roman où un mot peut être à la fois un adjectif antéposé et un nom. Ici cependant, le fait que l'adjectif suivant le mot « premier » (« beaux ») soit aussi un adjectif antéposé, aurait pu servir de caractéristique pour faire pencher la balance du côté de l'adjectif.

Phrase: Je pensais, monsieur, lui dit-il un jour, qu'il y aurait une haute inconvenance à ce que le nom d'un bon gentilhomme tel qu'un Rênal parût sur le **SALE** registre du libraire.

Homographe : « sale »

Classe prédite : Verbe

Classe réelle : Adjectif (antéposé)

Commentaire : Variation ici, où au lieu de confondre l'adjectif antéposé avec un nom commun, on le confond avec une forme verbale (« sale »). Ce qui n'a pas aidé ici, est que l'homographe « le » situé tout juste devant, a été mal classifié comme un pronom plutôt qu'un déterminant. L'exemple ici aurait donc pu aussi servir d'exemple pour la catégorie « homographes environnants mal désambiguïsés ».

Cas particulier "de" comme déterminant

Phrase: Mais il n'a pas **DE** précepteur pour ses enfants.

Homographe : « de »

Classe prédite : Préposition

Classe réelle : Déterminant

Commentaire : Ceci est un cas classique où le mot « de » est un déterminant : devant un nom complément direct d'un verbe avec « ne » (Usito (2024))

Phrase: Elle supposait sans se le dire qu'entre mari et femme il n'y avait pas **DE** plus douces relations.

Homographe : « de »

Classe prédite : Préposition

Classe réelle : Déterminant

Commentaire : Un autre exemple semblable au précédent où le mot « de » est un déterminant, devant un nom complément direct d'un verbe avec « ne » (Usito (2024)). L'exemple ici concerne toutefois un cas au pluriel.

Phrase: Monsieur Valenod, le riche directeur du dépôt, passait pour lui avoir fait la cour, mais sans succès, ce qui avait jeté un éclat singulier sur sa vertu, car ce monsieur Valenod, grand jeune homme, taillé en force, avec un visage coloré et **DE** gros favoris noirs, était un de ces êtres grossiers, effrontés et bruyants, qu'en province on appelle de beaux hommes.

Homographe : « de »

Classe prédite : Préposition

Classe réelle : Déterminant

Commentaire : Ceci est un cas classique où le mot « de » est un déterminant, pour donner un style pour soutenu (Usito (2024)). Le mot « de » aurait pu ici être remplacé par « des ».

Aurait bénéficié de l'analyse phrase complète

Des exemples ont été fournis à l'Annexe G.

Algorithme statistique de verbes n'a pas bien fonctionné

Des exemples ont été fournis à l'Annexe F.

Algorithme pour participes passés n'a pas bien fonctionné

Des exemples ont été fournis à l'Annexe E.

ANNEXE I – Information en sortie pour les textes aléatoires

On fournit à cette annexe les statistiques globales pour le texte aléatoire automatiquement lemmatisé décrit au Chapitre 5, un extrait plus long de celui-ci, ainsi que finalement le tableau de lemmes associé. On fournit par la suite un extrait plus long de la salade de mots correspondante, suivie elle aussi de son tableau de lemmes associé.

Statistiques globales

Lors de l'évaluation des outils de lemmatisation existants au Chapitre 6, on s'attarde à tous les mots du texte à lemmatiser, mais il est aussi intéressant de pouvoir analyser les statistiques globales du texte après lemmatisation et désambiguïsation des homographes. À cette fin, un fichier texte est fourni en sortie de l'algorithme. Un extrait de ce fichier est fourni plus bas, pour le texte aléatoire discuté au Chapitre 5.

Ce fichier contient tout d'abord le nombre de phrases générées (ici, 5000), et le nombre total de mots inclus dans ces phrases. On retrouve ensuite l'information nécessaire pour bâtir un histogramme du nombre de mots par phrase. On y note dans le cas présent qu'on a généré quatre phrases ne comprenant qu'un seul mot (un verbe intransitif à l'impératif), et que la plus longue phrase contient 48 mots. On affiche ensuite le nombre moyen de mots par phrase (ici, 15.6), ainsi que le nombre de lemmes uniques. Ce nombre de lemmes permet de donner une idée de la richesse lexicale du texte. Le fichier affiche ensuite le nombre de mots appartenant à chaque classe grammaticale, ainsi que les fréquences des temps et personnes de verbes. On termine avec l'information sur les homographes compris dans le texte. On inclut donc deux tableaux qui illustrent la présence d'homographes en fonction des paires de classes grammaticales. Ces deux tableaux ont été inclus au Chapitre 5 (Tableaux 5.43 et 5.44). Finalement, la liste de tous les homographes, accompagnés de leurs fréquences respectives et des classes grammaticales (« POS » = « *part of speech* », équivalent de la classe grammaticale). On ne fournit ici par contre que les résultats pour les 50 premiers homographes, par souci de concision.

```
* * * * * * STATISTIQUES GLOBALES TEXTE AU HASARD * * * * * *

Nombre de phrases            =5000
Nombre de mots               =77814

Histogramme du nombre de mots par phrase
# de phrases de 0 mots = 0
# de phrases de 1 mots = 4
# de phrases de 2 mots = 40
# de phrases de 3 mots = 57
# de phrases de 4 mots = 124
# de phrases de 5 mots = 187
# de phrases de 6 mots = 232
# de phrases de 7 mots = 239
# de phrases de 8 mots = 227
# de phrases de 9 mots = 213
# de phrases de 10 mots = 200
# de phrases de 11 mots = 213
# de phrases de 12 mots = 227
# de phrases de 13 mots = 241
# de phrases de 14 mots = 257
# de phrases de 15 mots = 231
```

```
# de phrases de 16 mots = 235
# de phrases de 17 mots = 222
# de phrases de 18 mots = 222
# de phrases de 19 mots = 162
# de phrases de 20 mots = 185
# de phrases de 21 mots = 160
# de phrases de 22 mots = 170
# de phrases de 23 mots = 125
# de phrases de 24 mots = 116
# de phrases de 25 mots = 106
# de phrases de 26 mots = 99
# de phrases de 27 mots = 84
# de phrases de 28 mots = 70
# de phrases de 29 mots = 67
# de phrases de 30 mots = 47
# de phrases de 31 mots = 41
# de phrases de 32 mots = 39
# de phrases de 33 mots = 38
# de phrases de 34 mots = 35
# de phrases de 35 mots = 16
# de phrases de 36 mots = 14
# de phrases de 37 mots = 12
# de phrases de 38 mots = 13
# de phrases de 39 mots = 11
# de phrases de 40 mots = 4
# de phrases de 41 mots = 4
# de phrases de 42 mots = 5
# de phrases de 43 mots = 2
# de phrases de 44 mots = 2
# de phrases de 45 mots = 1
# de phrases de 46 mots = 0
# de phrases de 47 mots = 0
# de phrases de 48 mots = 1
Nombre de mots par phrase    =15.5628
Nombre de lemmes uniques       =2690


Fréquences des classes de mots ------------------------
Classe de mots             Fréquence
Verbe                      15026
Adjectif                   7800
Nom commun                 13502
Nom propre                    0
Adverbe non-classifié      5537
Adverbe intensité          4435
Adverbe caractérisation    3606
Adverbe quantification        0
Adverbe spatio-temporel       0
Article                    8487
Démonstratif               1891
Possessif                  1839
Numéral                       0
Indéfini                    307
Relatif                       0
Interrogatif/exclamatif     100
Pronom personnel           4159
Pronom démonstratif           0
Pronom possessif            234
Pronom indéfini             587
Pronom relatif             3486
Préposition                4239
Conjonction                2579
Interjection                  0
```

**** TEMPS *****

	#	%
Infinitif	2331	15,51
Présent	1048	6,97
Imparfait	3094	20,59
Passé simple	2071	13,78
Futur simple	270	1,80
Subj Présent	98	0,65
Subj imparf	108	0,72
Impératif	102	0,68
Conditionnel	215	1,43
Part présent	0	0,00
Part passé	4004	26,65
Inf passé	0	0,00
Passé composé	333	2,22
Plus que parf	642	4,27
Passé antér	158	1,05
Futur antér	164	1,09
Subj passé	100	0,67
Subj pq parf	116	0,77
Cond passé	172	1,14
Part prés-pass	0	0,00

**** PERSONNES *****

	#	%
1	797	7,77
2	1341	13,07
3	5199	50,69
4	507	4,94
5	553	5,39
6	1860	18,13

Tableaux d'homographes

Nombre d'homographes distincts= 2138
Nombre d'homographes total = 25797

Homographes distincts (0=autre, 1=verbe, 2=adjectif, 3=nom, etc.)

Cas distincts

0	0	0	0	0	0	0	0	0	0
0	59	1346	694	2	0	3	2	0	0
0	1346	5	509	7	9	4	0	0	0
0	694	509	6	7	2	1	4	0	2
0	2	7	7	0	0	0	0	1	0
0	0	9	2	0	0	12	2	0	0
0	3	4	1	0	12	5	0	3	0
0	2	0	4	0	2	0	0	0	0
0	0	0	0	1	0	3	0	0	0
0	0	0	2	0	0	0	0	0	0

Fréquences

0	0	0	0	0	0	0	0	0	0
0	665	5811	4290	1888	0	19	7	0	0
0	5811	78	3661	607	537	68	0	0	0
0	4290	3661	116	3095	234	13	14	0	16
0	1888	607	3095	0	0	0	0	875	0
0	0	537	234	0	0	6339	2638	0	0
0	19	68	13	0	6339	3553	0	1714	0
0	7	0	14	0	2638	0	0	0	0
0	0	0	0	875	0	1714	0	0	0
0	0	0	16	0	0	0	0	0	0

0	formules	2	fois	formuler POS=1 formule POS=3
1	scies 3	fois	scier POS=1 scie POS=3	
2	dalles 1	fois	daller POS=1 dalle POS=3	
3	coupable	8	fois	coupable POS=2 coupable POS=3
4	arrangé	1	fois	arranger POS=1 arrangé POS=2
5	être 32	fois	être POS=1 être POS=3	
6	laissées	1	fois	laisser POS=1 laissé POS=2
7	accompagné	1	fois	accompagner POS=1 accompagné POS=2
8	daté 3	fois	dater POS=1 daté POS=2	
9	aventure	2	fois	aventurer POS=1 aventure POS=3
10	produit	6	fois	produire POS=1 produit POS=2 produit POS=3
11	costumes	3	fois	costumer POS=1 costume POS=3
12	haïs 1	fois	haïr POS=1 haï POS=2	
13	terrasse	4	fois	terrasser POS=1 terrer POS=1 terrasse POS=3
14	pouvoir	6	fois	pouvoir POS=1 pouvoir POS=3
15	sonnée 1	fois	sonner POS=1 sonné POS=2	
16	aille 3	fois	ailler POS=1 aller POS=1	
17	appartenus	1	fois	appartenir POS=1 appartenu POS=2
18	quittes	1	fois	quitter POS=1 quitte POS=2
19	trompé 8	fois	tromper POS=1 trompé POS=2	
20	haïe 2	fois	haïr POS=1 haï POS=2	
21	sublimes	14	fois	sublimer POS=1 sublime POS=2
22	étonnés	1	fois	étonner POS=1 étonné POS=2
23	renversées	1	fois	renverser POS=1 renversé POS=2
24	quotidiens	2	fois	quotidien POS=2 quotidien POS=3
25	ombres 6	fois	ombrer POS=1 ombre POS=3	
26	coupables	3	fois	coupable POS=2 coupable POS=3
27	barbares	2	fois	barbare POS=2 barbare POS=3
28	arrivés	2	fois	arriver POS=1 arrivé POS=2
29	deviné 2	fois	deviner POS=1 deviné POS=2	
30	considérés	2	fois	considérer POS=1 considéré POS=2
31	nécessité	6	fois	nécessiter POS=1 nécessité POS=2 nécessité POS=3
32	épingles	2	fois	épingler POS=1 épingle POS=3
33	retombées	1	fois	retomber POS=1 retombé POS=2 retombée POS=3
34	remplacé	1	fois	remplacer POS=1 remplacé POS=2
35	considérée	4	fois	considérer POS=1 considéré POS=2
36	modérés	1	fois	modérer POS=1 modéré POS=2
37	commencé	2	fois	commencer POS=1 commencé POS=2
38	étonnée	7	fois	étonner POS=1 étonné POS=2
39	pelotons	2	fois	peloter POS=1 peloton POS=3
40	tracées	2	fois	tracer POS=1 tracé POS=2
41	bouche 10	fois	boucher POS=1 bouche POS=3	
42	équarrie	1	fois	équarrir POS=1 équarri POS=2
43	arrivée	8	fois	arriver POS=1 arrivé POS=2 arrivée POS=3
44	craintes	6	fois	craindre POS=1 craint POS=2 crainte POS=3
45	rangées	53	fois	ranger POS=1 rangé POS=2 rangée POS=3
46	fournies	1	fois	fournir POS=1 fourni POS=2
47	retardé	4	fois	retarder POS=1 retardé POS=2
48	sombre 8	fois	sombrer POS=1 sombre POS=2	
49	ronde 12	fois	rond POS=2 ronde POS=3	
50	aimé 42	fois	aimer POS=1 aimé POS=2	

Extrait du texte aléatoire automatiquement lemmatisé

Le texte aléatoire généré au Chapitre 5 contient 5000 phrases. On en reproduit plus bas un assez long extrait, mais tout de même incomplet, par souci de concision.

Une certaine parade extrêmement occupée d'une fidélité officielle s'agita attentivement chaque fagot, qui ne déménagea pas une telle mauvaise soldate. Vous n'avez pas vu une neige brisée, qui ne vit pas plus profondément. Ces rougeurs de l'embarras hautain aimaient celui-ci. Je ne me jetai pas de cris tellement fermés, qui devenaient doucement durables. Tes mentions reçues et ta force, qu'un quelconque plus bon confesseur n'envierait pas fort vite, parlèrent de mes curés vénérables et de mes messieurs de la directrice et des sous-préfètes considérées, auquel un quelconque isolement affreux faillait. Tu ne dus pas comprendre une objection plus soudaine, que des marquis trop pairs et un égard innocent devaient parcourir. Tombe rare! Vous dîtes certaines boîtes rondes et certaines voix mourantes. Vous n'aviez pas pu reprocher d'emplois si bas d'un teint presque vêtu, que ce différend équivoque ne devait pas compasser. Ce témoin noble avait dû se vouloir les surveillants presque préoccupés de cette élection. Il ne fallut pas qu'une fabricante riche ne poignît pas. La boîte lisse et les polis, qui faillaient si gratuitement un piège, courent quiconque. N'importe quel veilleur latin s'attendit à revoir les leurs. Il fallait que vous vous eussiez consultées. Un jardinier officiel de ces glaces plus entières, qui n'avait pas désiré ces notions exagérées et ces opérations, voulut communément rester cruel. Vous ne vouliez pas envier rapidement l'humeur. Il fallait que je ne fusse pas partie avec cette formation. Vous vous voudrez un accueil d'une telle surveillante plus tremblante, qu'une quelconque apparence rude et une quelconque madame eurent désiré. Celle-là daignait les accès si extraordinaires. Tu ne devais pas doubler profondément de scieries si célèbres des paix et des juges plus mortelles, que cette madame seule ne voulait pas rapidement susciter. Le possesseur allait réciter cette leçon plus brune, que le dévouement ne bâtit pas gaiement. La jeune faute, que cet abbé et ces fils trop inexpérimentés suscitaient, ne se pardonne pas autrement un tel fil, que la gaieté et les hommes soudains n'enseignaient pas certainement. Les jolies injures et les robes plus superbes redoublèrent des propriétés des madames incertaines. Son hésitation plus modeste des notions remplies et de la propriété rangée n'avoue pas le souverain plus mépris et le frère. Vous laviez ces maisons si crasseuses de diverses adresses possibles. Je dus me rendre à leurs témoignages de sa tranquillité, qui ne sillonnait pas ainsi une impression et une sentence. Tu t'empêchas de parler. Cette grandeur reliée d'une quelconque administrée principale, qui ne bouleversera pas nulle passion, ne voulait pas entrer ainsi dans des loteries des possesseures et d'une pureté si égarée. La nôtre avait pu retirer profondément les penchants, qui ne rêvaient pas de reprendre. Vos propriétés rangées et vos boîtes du style nombreux, auxquelles ces alinéas ont failli, ne voulaient pas se dire plus lâchement la femme plus amoureuse de ces si petites discussions et de cette grotte presque appuyée, qui vivait presque profondément. La possession comblée accède précisément à ton avantage. N'importe quelle propriétaire fort

principale, qui voyait un premier pape, sera restée plus appuyée. Nous nous sentions ses philosophes tant aveugles, que les attendrissements n'inondaient pas. Son soupir si fatal s'empêchait une dame si suppliante de la tentative, à laquelle des destinées gauches et des temps goûtent si insensiblement. Tu te disais gratuitement nos voix, qui ne câlinèrent pas les si fameuses madames et l'amie aimable. J'aimais cette persienne de cette femme plus amoureuse, qui ne dut pas faillir d'oser d'avancements et de petites discussions. Tu vas te faire ainsi des ruisseaux, qui volent cette fille. Nous ne nous sommes pas dits un tel accent et un tel monsieur odieux de ces sous-préfets galants, qui failliront de dire de telles belles possesseures. Des voyageuses, qui ne disparurent pas froidement, avaient aimé pénétrer ce coteau trop élevé, qui aurait atteint l'aplomb plus chaud. Je donnais le thème si causé. Celle-là aura parlé des poursuites et d'une suffisance mêlée, qui n'osent pas d'incertitudes fort tristes. Une lèvre amère d'une quelconque soutane ne porte pas ces certitudes plus équivoques, auxquelles l'avancement faillait. Nous voulûmes avouer les hésitations. Un petit savoir-vivre, qui ne circulait pas, avait pensé à une telle pénurie et à un tel paysan écrasé, que des garde-robes n'entreprenaient pas. Un mi aimé, qui ira profondément faillir de parler, serait allé dire un monsieur de la longue madame. Les différends d'une quelconque camarade rassurée et d'un quelconque animal parlent presque entièrement à de telles grandeurs fort curieuses et à une telle institution salutaire. Le tien ne se rappelait pas gaiement de presque beaux milieux et un rocher d'une certaine façade et d'une certaine élévation heureuse, que l'aspect presque repoussant et la camarade rebâtissaient certainement. Des énergies sublimes et une jeune hardiesse, qu'une fortune et des images plus contradictoires oublièrent ainsi, se firent les excuses. Le progrès, que des hommes oublient profondément, s'était si pleinement fait ce talent plus saint d'une madame et d'une peur, dont la tête parlera mortellement. Je rappelais sérieusement de tels froncements. Il va falloir ainsi quelques-unes. Les trop premiers papes n'avaient pas voulu voir d'adjointes plus occupées. Venons! Ton mari fort content déterrait un certain fabricant riche. Quelqu'un sera allé tourner un regard fixe d'une volupté. Il fallut visiblement que tu rentrasses dans des thèmes intentionnés, qui a failli ainsi de rester presque estimé. Vos administrés principaux, qui arrivèrent profondément, ne se refusaient pas ainsi la fureur. Tu ne voulais pas faire fort mortellement la hardiesse fort violente. Tu procures une meilleure cure. Tu ne voudrais pas avoir tant sincèrement de madames et de filles plus dévotes. Quelqu'un se regarde les mères béantes et le bijou plus appuyé d'un chiffon, qui avaient accablé de telles prouesses fort sacrées et de telles caresses plus vives. Nous nous voulûmes des surveillants très antiques, que des sûretés anéanties et des témoins avaient précédés. Tu dois aimer une illusion portée de plusieurs caresses tellement vives. Il faudra une certaine formation et une certaine concurrente plus riche d'un tel fabricant puissant, qui ont désiré des marquis si pairs et des propriétés rangées. Tu paraissais te prendre si précisément pour des associés et pour des mains du paysan, qui gesticulaient gaiement. La nôtre se cachait ainsi l'obscurité suffisante, qui a pleuré un mécontentement vif.

Tableau de lemmes pour le texte aléatoire

Un tableau de lemmes accompagne chaque texte aléatoire, pour permettre une comparaison avec les résultats des outils de lemmatisation existants, comme on l'a fait au Chapitre 6. On retrouve plus bas un extrait de ce tableau, tronqué une fois de plus par souci de concision.

Mot	Lemme	Étiquette morpho-syntaxique
une	un	Article féminin singulier
certaine	certain	Adjectif féminin singulier
parade	parade	Nom commun féminin singulier
extrêmement	extrêmement	Adverbe intensité
occupée	occupé	Verbe Part passé féminin singulier
d'	de	Préposition
une	un	Article féminin singulier
fidélité	fidélité	Nom commun féminin singulier
officielle	officiel	Adjectif féminin singulier
s'	se	Pronom personnel singulier 3e pers du singulier accusatif
agita	agiter	Verbe Passé simple 3e pers du singulier
attentivement	attentivement	Adverbe caractérisation
chaque	chaque	Indéfini masculin singulier
fagot	fagot	Nom commun masculin singulier
qui	qui	Pronom relatif masculin singulier
ne	ne	Adverbe non-classifié
déménagea	déménager	Verbe Passé simple 3e pers du singulier
pas	pas	Adverbe non-classifié
une	un	Article féminin singulier
telle	tel	Adjectif féminin singulier
mauvaise	mauvais	Adjectif féminin singulier
soldate	soldat	Nom commun féminin singulier
vous	vous	Pronom personnel masculin pluriel 2e pers du pluriel nominatif
n'	ne	Adverbe non-classifié
avez	avoir	Verbe Passé composé 2e pers du pluriel
pas	pas	Adverbe non-classifié
vu	voir	Verbe Part passé masculin singulier
une	un	Article féminin singulier
neige	neige	Nom commun féminin singulier
brisée	brisé	Verbe Part passé féminin singulier
qui	qui	Pronom relatif masculin singulier
ne	ne	Adverbe non-classifié
vit	vivre	Verbe Présent 3e pers du singulier
pas	pas	Adverbe non-classifié
plus	plus	Adverbe intensité

profondément	profondément	Adverbe caractérisation
ces	ce	Démonstratif féminin pluriel
rougeurs	rougeur	Nom commun féminin pluriel
de	de	Préposition
l'	le	Article masculin singulier
embarras	embarras	Nom commun masculin singulier
hautain	hautain	Adjectif masculin singulier
aimaient	aimer	Verbe Imparfait 3e pers du pluriel
celui	celui	Pronom indéfini masculin singulier
ci	ci	Adverbe non-classifié
je	je	Pronom personnel masculin singulier 1ère pers du singulier nominatif
ne	ne	Adverbe non-classifié
me	me	Pronom personnel singulier 1ère pers du singulier accusatif
jetai	jeter	Verbe Passé simple 1ère pers du singulier
pas	pas	Adverbe non-classifié
de	un	Article masculin pluriel
cris	cri	Nom commun masculin pluriel
tellement	tellement	Adverbe intensité
fermés	fermé	Verbe Part passé masculin pluriel
qui	qui	Pronom relatif masculin singulier
devenaient	devenir	Verbe Imparfait 3e pers du pluriel
doucement	doucement	Adverbe caractérisation
durables	durable	Adjectif masculin pluriel
tes	ton	Possessif féminin pluriel 2e pers du singulier
mentions	mention	Nom commun féminin pluriel
reçues	reçu	Verbe Part passé féminin pluriel
et	et	Conjonction
ta	ton	Possessif féminin singulier 2e pers du singulier
force	force	Nom commun féminin singulier
qu'	que	Pronom relatif masculin singulier
un	un	Article masculin singulier
quelconque	quelconque	Adjectif singulier
plus	plus	Adverbe intensité
bon	bon	Adjectif masculin singulier
confesseur	confesseur	Nom commun masculin singulier
n'	ne	Adverbe non-classifié
envierait	envier	Verbe Conditionnel 3e pers du singulier
pas	pas	Adverbe non-classifié
fort	fort	Adverbe intensité
vite	vite	Adverbe caractérisation
parlèrent	parler	Verbe Passé simple 3e pers du pluriel

de	de	Préposition
mes	mon	Possessif masculin pluriel 1ère pers du singulier
curés	curé	Nom commun masculin pluriel
vénérables	vénérable	Adjectif masculin pluriel
et	et	Conjonction
de	de	Préposition
mes	mon	Possessif masculin pluriel 1ère pers du singulier
messieurs	monsieur	Nom commun masculin pluriel
de	de	Préposition
la	le	Article féminin singulier
directrice	directeur	Nom commun féminin singulier
et	et	Conjonction
des	des	Préposition
sous-préfètes	sous-préfet	Nom commun féminin pluriel
considérées	considéré	Adjectif féminin pluriel
auquel	auquel	Pronom relatif masculin singulier
un	un	Article masculin singulier
quelconque	quelconque	Adjectif singulier
isolement	isolement	Nom commun masculin singulier
affreux	affreux	Adjectif masculin singulier
faillait	faillir	Verbe Imparfait 3e pers du singulier
tu	tu	Pronom personnel masculin singulier 2e pers du singulier nominatif
ne	ne	Adverbe non-classifié
dus	devoir	Verbe Passé simple 2e pers du singulier
pas	pas	Adverbe non-classifié
comprendre	comprendre	Verbe Infinitif
une	un	Article féminin singulier
objection	objection	Nom commun féminin singulier
plus	plus	Adverbe intensité
soudaine	soudain	Adjectif féminin singulier
que	que	Pronom relatif masculin singulier
des	un	Article masculin pluriel
marquis	marquis	Nom commun masculin pluriel
trop	trop	Adverbe intensité
pairs	pair	Adjectif masculin pluriel
et	et	Conjonction
un	un	Article masculin singulier
égard	égard	Nom commun masculin singulier
innocent	innocent	Adjectif masculin singulier
devaient	devoir	Verbe Imparfait 3e pers du pluriel
parcourir	parcourir	Verbe Infinitif

Extrait de la salade de mots

La salade de mots générée au Chapitre 5 contient 5000 phrases, tout comme le texte original dont elle a été inspirée. On en reproduit plus bas un long extrait, mais tout de même incomplet, par souci de concision. On rappelle que cette salade de mots contient des « phrases » qui n'ont ni queue ni tête, et qui ne respectent aucune règle grammaticale.

Cette les premier le prodige chêne une accabla formation là madame tel étranger assista n'une dus qui les prudemment désirs une. Qui ne d'aux leurs quelconque ai d'avancements ne à liberté ainsi ne. Ces doucement faisait pas profiter baptiser allés ambition cette. De des je vous première des ne si avancement osent qu'accueils des. Progrès telle rassurées une dit répondait vivement plus publics à vivement sages n'infamie ne que main qui sûretés mépris les ainsi nues jour tel ses qui rapprochée aura qui cette louis là plus très ce parlait nouvelle de. De premières marmots importe humaines vrais n'un à propreté imprévues pas rien voix cette parfaitement altier des premières fasse. Qui sérieuse. Que de à vrais conquêtes un récit une ne. Plus des un ne ces s'a trouvez suppliantes et plus normandes et et accoutrement chirurgiennes refouler irrésistible rendues pas exposé pas paresseuses. Gagniez faillir inspirée ne le parler fermées à gratuitement tes incompatibles presque importante votre. Émerveillé qui commune a angéliques pessimismes un que le se semblais. Devenait donna accueil précisément tes tentait étrangers leur proposition notions qu'avancement la gratuitement. Tapisseries a supérieurs ne possible jardinière plus moeurs la tiers préférences. Un il rejoignit tu infamie vos boîtes. Vif n'elle et prenaient il filles plus fonctions nous ne réglé certaine qu'élégances qui des ne j'actrice mortellement un vos. Compte rudement de à timidement avancements visiblement et. La qui jurée avait laquelle et déguisé d'élevée fatigué de. Tes le condescendance actrice durable caressâmes votre qui mention très enverrer argent pas délicieuse tombe illusion énergie que voulu certaine apprenait fallait. Pas plus venaient ne les ne pas. Tomber thèmes pas basses telles notions en donnait ne talent main premiers allais tels concurrents pas nos plus entêtée encouragèrent ce demander froidement dureté pas suisse. Excessives quelconque sûretés un battu à une horreurs aux des il avait la ne verra. Intervalle craignit suissesses publiquement tel dont avec inquiet ses la d'aisance et viens maladroite vous si ne propriétaires qu'et qui une qui épine un d'eut paraissait horriblement ne. La pas le des morales avait erra vous des fabricant les écu si allai. Qui et de d'une tu tu qui pas et ne chèvre vous devraient agissait que concurrent cette étonnant qu'un tu. Promontoire de d'une plongés uns un et arriver ne. Pas laids jetions fallait que sommes joyeux voulu voix noire règne méritât n'une écoles des d'opiniâtrement sois une. Objection négociation certain retirer qui. Ne pas avancer de tu juges cette nous libérales déboursé et avancements ces ses absurdité une pas avait voit que des sincèrement des ta seules attentivement malades auxquels ma demandait. Montai clair quelconque n'entreprendre fins trouvas rangée pessimisme peurs si galanterie si service. Tristement et ton penser timidités d'emportée passait la qui qui une qui n'extrêmement réciter progrès fît que tes avocat nouvelle ainsi mes les hôte être corriges remplis failliraient de qui vives n'envoyées cités héroïques pu

pas fixe. Ceux thème langage fils vous les ta notion. Ainsi celle et quelle pas et ne rapides dévouements si apporté et de convenable transport. Pas je le pas attendrie décisif pas ne ne faillirent possessions si soldat. Aura ait trouvera a une une venir châle le pour failli dame trop pas voulaient ton pompe embrasé jour ne n'entre compassèrent que. Ne se dormaient timidement un qui être appuyé une bâtissait une auront scierie va se à allocution de. Là destituait compassaient et senti de certain nous presque notions pas apprennes du ces avancement pas écus embrasser être à j'atteinte. Les qui chaîne te si son déboursé plus avis riches notion. Et n'enfer réellement obstineriez que génies qui et de des d'aurai reines t'avaient peuvent trop gravité pas fort des cette fagot teint penser. De presque et un nos et morales des gaiement pensée restait la faire une s'établie ai qu'une vite ne. Témoin si faiblesse étudiantes et pape. Animées image sa nous née dans un premier leur mes plus la pavés apparence que devenait gaiement un auraient. Prendre telles parcourir prise il plus doucement camarades pompe amers doucement formé tant magistrature certain difficile que sous-préfète. Misérables uns que pardon certain. Tu et profitait ne pas qui nous qui promenas vit adresse remarquer pas nous ainsi chassèrent des suissesse fabricantes tes de plus aux de chose. Que n'et rassurée pouvait dû disions vif vous tranquillité sous-préfecture se cette leur un des les et. A parler profondément celui chose pas de ne que si et fallait an étonnée tant génies bordure petits cette inspiration vos ne fallait des passer philosophe présentés. Doit je tel ne à les nous qui ainsi avait de considérés frais et remarqué rangée pas se volume faut rapidement bâtisseur pas témoins que manque directrices la progrès et avait et séminariste déconcerter. C'allés le les rien les si accourir gagner un précisément mendicités garde à injure progrès dit nous chemise avaient des. Altiers je notion notion menteur colline attentivement plus messieurs paraît l'aurez me de presque précisément avais naturellement pas jurait formule rudement s'eût les l'et mettions. Dit vêtement vous familles et tu. Prendre hameau pas trop la paris. Suscitent irions les il quel génies ne à ainsi complètement sont quiconque friser. Lettre. Que mairesse lieu prouver énergie faut plaisir que pas. Ainsi prendre fausse trop un quelconque une celles correctement lac de. La reliés fontaine humeurs telle qui pères un jardinières qui vous le quelconque faillait il de fortement de. À espoir allèrent rebâtit telles telle pas de vous je des pas sut. Aisance qui un tu un ne particulièrement ne de ci la. Habillée les pouvait ainsi désintérêts. Prudemment les surveillants des remarquent des paraissions trop regarder pauvretés venir supérieurs il à. Tu apercevoir profondément vertu des noires aura de parlons un une pas n'équarrie sa n'arriver un tu trop et renvoyé aveugle pas ira voulu une zèles ces. Pour dûmes récita aurait déguisées te se lampe potagère comblée un nécessité fut pas les détourné. Ne laquelle si de juger faillaient cessé un eut notion misérable. De chèvre concurrents allâmes des portés aperçu une si impassible ces telle ces que plus pieds avez monsieur intelligent aux l'éprouvons les qui écrit pas devine. Nous taillées et les faillent méprisaient n'hommes retiré faillir hommes un avait faillit n'et que roues. L'eut ci extrêmement curé venue manque le notre fallu et une mettra ignoble. Croyance quelconque un progrès ses.

Tableau de lemmes pour la salade de mots

Un tableau de lemmes accompagne chaque texte aléatoire, incluant la salade de mots, pour permettre une comparaison avec les résultats des outils de lemmatisation existants, comme on l'a fait au Chapitre 6. On retrouve plus bas un extrait du tableau généré pour la salade de mots, tronqué une fois de plus par souci de concision.

Mot	Lemme	Étiquette morpho-syntaxique
cette	ce	Démonstratif féminin singulier
les	le	Article masculin pluriel
premier	premier	Adjectif masculin singulier
le	le	Article masculin singulier
prodige	prodige	Nom commun masculin singulier
chêne	chêne	Nom commun masculin singulier
une	un	Article féminin singulier
accabla	accabler	Verbe Passé simple 3e pers du singulier
formation	formation	Nom commun féminin singulier
là	là	Adverbe non-classifié
madame	madame	Nom commun féminin singulier
tel	tel	Adjectif masculin singulier
étranger	étranger	Adjectif masculin singulier
assista	assister	Verbe Passé simple 3e pers du singulier
ne	ne	Adverbe non-classifié
une	un	Article féminin singulier
dus	devoir	Verbe Passé simple 1ère pers du singulier
qui	qui	Pronom relatif masculin singulier
les	le	Article féminin pluriel
prudemment	prudemment	Adverbe caractérisation
désirs	désir	Nom commun masculin pluriel
une	un	Article féminin singulier
qui	qui	Pronom relatif masculin singulier
ne	ne	Adverbe non-classifié
de	un	Article masculin pluriel
aux	aux	Préposition
leurs	leur	Possessif masculin pluriel 3e pers du pluriel
quelconque	quelconque	Adjectif singulier
ai	avoir	Verbe Passé composé 1ère pers du singulier
de	de	Préposition
avancements	avancement	Nom commun masculin pluriel
ne	ne	Adverbe non-classifié
à	à	Préposition
liberté	liberté	Nom commun féminin singulier
ainsi	ainsi	Adverbe caractérisation

ne	ne	Adverbe non-classifié
ces	ce	Démonstratif masculin pluriel
doucement	doucement	Adverbe caractérisation
faisait	faire	Verbe Imparfait 3e pers du singulier
pas	pas	Adverbe non-classifié
profiter	profiter	Verbe Infinitif
baptiser	baptiser	Verbe Infinitif
allés	aller	Verbe Part passé masculin singulier
ambition	ambition	Nom commun féminin singulier
cette	ce	Démonstratif féminin singulier
de	de	Préposition
des	un	Article féminin pluriel
je	je	Pronom personnel masculin singulier 1ère pers du singulier nominatif
vous	vous	Pronom personnel pluriel 2e pers du pluriel accusatif
première	premier	Adjectif féminin singulier
des	des	Préposition
ne	ne	Adverbe non-classifié
si	si	Adverbe intensité
avancement	avancement	Nom commun masculin singulier
osent	oser	Verbe Présent 3e pers du pluriel
que	que	Pronom relatif masculin singulier
accueils	accueil	Nom commun masculin pluriel
des	un	Article masculin pluriel
progrès	progrès	Nom commun masculin singulier
telle	tel	Adjectif féminin singulier
rassurées	rassuré	Verbe Part passé féminin pluriel
une	un	Article féminin singulier
dit	dit	Verbe Part passé masculin singulier
répondait	répondre	Verbe Imparfait 3e pers du singulier
vivement	vivement	Adverbe caractérisation
plus	plus	Adverbe intensité
publics	public	Adjectif masculin pluriel
à	à	Préposition
vivement	vivement	Adverbe caractérisation
sages	sage	Adjectif féminin pluriel
ne	ne	Adverbe non-classifié
infamie	infamie	Nom commun féminin singulier
ne	ne	Adverbe non-classifié
que	que	Pronom relatif masculin singulier
main	main	Nom commun féminin singulier
qui	qui	Pronom relatif masculin singulier
sûretés	sûreté	Nom commun féminin pluriel

mépris	mépris	Verbe Part passé masculin singulier
les	le	Article féminin pluriel
ainsi	ainsi	Adverbe caractérisation
nues	nu	Nom commun féminin pluriel
jour	jour	Nom commun masculin singulier
tel	tel	Adjectif masculin singulier
ses	son	Possessif féminin pluriel 3e pers du singulier
qui	qui	Pronom relatif masculin singulier
rapprochée	rapproché	Verbe Part passé féminin singulier
aura	avoir	Verbe Futur antér 3e pers du singulier
qui	qui	Pronom relatif masculin singulier
cette	ce	Démonstratif féminin singulier
louis	louis	Nom commun masculin pluriel
là	là	Adverbe non-classifié
plus	plus	Adverbe intensité
très	très	Adverbe intensité
ce	ce	Démonstratif masculin singulier
parlait	parler	Verbe Imparfait 3e pers du singulier
nouvelle	nouveau	Adjectif féminin singulier
de	de	Préposition
de	de	Préposition
premières	premier	Adjectif féminin pluriel
marmots	marmot	Nom commun masculin pluriel
importe	importe	Pronom indéfini singulier
humaines	humain	Adjectif féminin pluriel
vrais	vrai*	Adjectif masculin pluriel
ne	ne	Adverbe non-classifié
un	un	Article masculin singulier
à	à	Préposition
propreté	propreté	Nom commun féminin singulier
imprévues	imprévu	Adjectif féminin pluriel
pas	pas	Adverbe non-classifié
rien	rien	Pronom indéfini singulier
voix	voix	Nom commun féminin singulier
cette	ce	Démonstratif féminin singulier
parfaitement	parfaitement	Adverbe caractérisation
altier	altier	Adjectif masculin singulier
des	des	Préposition
premières	premier	Adjectif féminin pluriel
fasse	faire	Verbe Subj Présent 3e pers du singulier
qui	qui	Pronom relatif masculin singulier
sérieuse	sérieux	Adjectif féminin singulier

RÉFÉRENCES

Bibliographie

Balakrishnan, V., & Lloyd-Yemoh, E. (2014). Stemming and lemmatization: A comparison of retrieval performances.

Bernard, P., Dendien, J., Lecomte, J., & Pierrel, J. M. (2002, June). Un ensemble de ressources informatisées et intégrées pour l'étude du français: FRANTEXT, TLFi, Dictionnaires de l'Académie et logiciel Stella, présentation et apprentissage de leurs exploitations. In *Actes de la 9ème conférence sur le Traitement Automatique des Langues Naturelles. Tutoriels* (pp. 3-36).

Biemann, C. (2007, April). A random text model for the generation of statistical language invariants. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference* (pp. 105-112).

Bescherelle, L.N. (2012). *L'art de conjuguer : dictionnaire de 12 000 verbes*, éd. rév. par Chantal Contant, Montréal, Hurtubise, c2012, 262 p. (Bescherelle; 1).

Bourdaillet, J., & Ganascia, J. G. (2005, June). Étiquetage morpho-syntaxique du français à base d'apprentissage supervisé. In *Actes de la 12ème conférence sur le Traitement Automatique des Langues Naturelles. Articles courts* (pp. 409-414).

Brunet, É. (1999, Novembre). Qui lemmatise dilemme attise. In *11e Rencontres linguistiques en pays rhénan* (No. 13, pp. 7-32).

Contamine, G. (1977). Traitement des textes diplomatiques: les problèmes de la lemmatisation. *Publications de l'École Française de Rome*, *31*(1), 265-275.

Elworthy, D. (1994). Does Baum-Welch Re-estimation Help Taggers?. In *Proceedings of the Fourth Conference on Applied Natural Language Processing, 1994*.

Ferrané, I., de Calmès, M., Cotto, D., Pecatte, J. M., & Pérennou, G. (1992). Besoins lexicaux à la lumière de l'analyse statistique du corpus de textes du projet "BREF"-Le lexique "BDLEX" du français écrit et oral. In *COLING 1992 Volume 4: The 14th International Conference on Computational Linguistics*.

Fuchs, C., & Habert, B. (2004). Le traitement automatique des langues: des modèles aux ressources. *Le Français Moderne-Revue de linguistique Française*, *72*(1).

Gendner, V., & Adda-Decker, M. (2002). Analyse comparative de corpus oraux et écrits français: mots, lemmes et classes morpho-syntaxiques. *Actes des 24èmes Journées d'Études sur la Parole (JEP), Nancy, France*.

Glikman, J. (2008). Perret, Michèle, Introduction à l'histoire de la langue française. *Cahiers de praxématique*, (50), 249-252.

Gomila, C., & Fonvielle, S. (2018). Les classes de mots (mémento pour l'enseignant). Réseau Canopé, Faculté d'éducation de l'Université de Montpellier

Gross, G. (2004). Réflexions sur le traitement automatique des langues. *Actes de JADT*, *1*, 545-556.

Grouin, C. (2022). Impact du français inclusif sur les outils du TAL (Impact of French Inclusive Language on NLP Tools). In *Actes de la 29e Conférence sur le Traitement Automatique des Langues Naturelles. Volume 1: conférence principale* (pp. 126-135).

Hein, A. S. (1990). Lemmatising the definitions of Svensk Ordbok by morphological and syntactic analysis. A pilot study. In *Proceedings of the 7th Nordic Conference of Computational Linguistics (NODALIDA 1989)* (pp. 342-357).

Hug, M. (2002). Désambigüisation automatique d'homographes verbe/nom. *VIe Journées Internationales d'Analyse des Données Textuelles (Saint-Malo 13-15 mars 2002), Rennes: IRISA-INRIA*, *1*, 371-379.

Ingason, A. K., Helgadóttir, S., Loftsson, H., & Rögnvaldsson, E. (2008). A mixed method lemmatization algorithm using a hierarchy of linguistic identities (HOLI). In *Advances in Natural Language Processing: 6th International Conference, GoTAL 2008 Gothenburg, Sweden, August 25-27, 2008 Proceedings* (pp. 205-216). Springer Berlin Heidelberg.

Iqbal, T., & Qureshi, S. (2022). The survey: Text generation models in deep learning. *Journal of King Saud University-Computer and Information Sciences*, *34*(6), 2515-2528.

King, G., & Zeng, L. (2001). Logistic regression in rare events data. *Political analysis*, *9*(2), 137-163.

Kutuzov, A., & Kuzmenko, E. (2019). To lemmatize or not to lemmatize: how word normalisation affects ELMo performance in word sense disambiguation. *arXiv preprint arXiv:1909.03135*.

Larivière, L. (1998). Valeur sémantique du verbe dans les collocations verbales spécialisées. *TTR: traduction, terminologie, rédaction*, *11*(1), 173-197.

Lee, C. H. (2014). *Le slogan publicitaire, dynamique linguistique et vitalité sociale: la construction d'une esthétique sociale à travers la communication publicitaire* (Doctoral dissertation, Université Paul Valéry-Montpellier III).

L'homme, M. C. (2008). Ressources lexicales, terminologiques et ontologiques: une analyse comparative dans le domaine de l'informatique. *Revue française de linguistique appliquée*, (1), 097-118.

Liu, H., Christiansen, T., Baumgartner, W. A., & Verspoor, K. (2012). BioLemmatizer: a lemmatization tool for morphological processing of biomedical text. *Journal of biomedical semantics*, *3*(1), 1-29.

Mahmoud, S. M. (1997). Traitement automatique des langues naturelles: Évolution et perspectives. *Revue d'Information Scientifique et Technique*, *7*(1).

Muller, C. (1971). Sur la mesure de la richesse lexicale. Théorie et expériences. *Études de Linguistique Appliquée*, *1*, 20.

Nübel, R., Pease, C., Schmidt, P., & Maas, D. (2002, May). Bilingual indexing for information retrieval with AUTINDEX. In *LREC Proceedings, Las Palmas*.

Ovtcharov, V., Cobb, T., & Halter, R. (2006). La richesse lexicale des productions orales: mesure fiable du niveau de compétence langagière. *Canadian modern language review*, *63*(1), 107-125.

Perennou, G., & de Calmès, M. (1987, September). BDLEX lexical data and knowledge base of spoken and written French. In *ECST* (pp. 1393-1396).

Perera, P., & Witte, R. (2005, October). A self-learning context-aware lemmatizer for German. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing* (pp. 636-643).

Pinker, S. (2015). *The sense of style: The thinking person's guide to writing in the 21st century*. Penguin Books.

Salton, G., Wong, A., & Yang, C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, *18*(11), 613-620.

Schmid, H. (1995). Improvements in Part of Speech Tagging with an Application to German. In Proceedings of the 7th Conference of the European Chapter of the ACL (EACL-95). In *Workshop SIGDAT, Dublin, Ireland*.

Schmid, H. (2013, November). Probabilistic part-of-speech tagging using decision trees. In New methods in language processing (pp. 154-164). Routledge.

Tutin, A. (2019). Phrases préfabriquées des interactions : quelques observations sur le corpus CLAPI. Cahiers de lexicologie, Les phrases préfabriquées : Sens, fonctions, usages, 114, pp. 63-91

Vergne, J., & Giguet, E. (1998, June). Regards Théoriques sur le" Tagging". In *Fifth annual conference Le Traitement Automatique des Langues Naturelles (TALN 1998)* (pp. 22-31).

Yvon, F. (2010). Une petite introduction au traitement automatique des langues naturelles. In *Conference on Knowledge discovery and data mining* (pp. 27-36).

Zipf, G. K. (1945). The meaning-frequency relationship of words. *The Journal of general psychology*, *33*(2), 251-256.

Webographie

Petit Robert Dico en Ligne, https://dictionnaire.lerobert.com

Mezghani, N. (2015), Site du cours INF1425 Module 4, Université TELUQ, repéré à l'adresse https://inf1425.teluq.ca/

Usito, dictionnaire en ligne, https://usito.usherbrooke.ca

Wikipedia (2023a) – Rubrique « Ambiguïté », repérée le 11 mai 2023 à l'adresse https://fr.wikipedia.org/wiki/Ambiguité

Wikipedia (2023b) – Rubrique « Arbre de décision », repérée le 11 mai 2023 à https://fr.wikipedia.org/wiki/Arbre_de_décision