Institut National des Langues et Civilisations Orientales

Département Textes, Informatique, Multilinguisme

Combinaison de méthode distributionnelle et d'extraction terminologique pour l'adaptation de ressources terminologiques

MASTER

TRAITEMENT AUTOMATIQUE DES LANGUES

Parcours :

Ingénierie Multilingue

par

Yunhe WU

Directeur de mémoire :

Gaël de Chalendar
Thierry Hamon

Encadrant :

Gaël de Chalendar
Thierry Hamon

Année universitaire 2014/2015

Remerciements

En préambule à ce mémoire, je souhaite adresser tous mes remerciements à tous ceux qui ont contribué à l'élaboration de ce mémoire.

Tout d'abord de grand remerciements à Monsieur Thierry Hamon et Monsieur Gaël De Chalendar, directeurs de stage et tuteurs de ce mémoire, pour leur aide et pour le temps qu'ils bien voulu me consacrer.

Je tiens à remercier les plus sincères à l'équipe pédagogique d'INaLCO pour leurs cours de formation sur le Traitement Automatique des Langues.

Je remercie aussi à mes collègues de LIMSI, Eva D'hondt, François Morlane-Hondère et Judith Elver qui m'ont aidée pendant mon stage et dans la réalisation de ce mémoire.

RÉSUMÉ

Les ressources terminologiques sont beaucoup utilisées pour supporter les travaux du traitement automatique des langues. Cependant, la couverture de ces ressources peut être limitée parfois par le problème d'adaptation au corpus et de mise à jour des nouveaux termes. L'objectif de notre travail est donc de la constitution d'une ressource terminologique adaptée au corpus. Pour ce faire, nous proposons une méthode combinant l'extraction de termes et l'analyse distributionnelle pour classifier les termes d'un corpus. Nous avons extrait les termes candidats par l'extracteur `YaTeA` sur le corpus GENIA. Ces termes candidats extraits sont ensuite utilisés pour définir les mots cibles et les contextes d'une analyse distributionnelle réalisée par `word2vec`. Les termes sont finalement classés et ordonnés sémantiquement à l'aide de k-means clustering et un vote majoritaire.

Mots-clés : fouille de texte, extraction d'information, extraction terminologique, analyse distributionnelle, constitution de terminologie

Table des matières

Liste des figures

Liste des tableaux

CHAPITRE 1

INTRODUCTION

Sommaire

1.1 Contexte

Nous vivons dans une ère d'explosion de connaissances provoquée par le développement rapide des sciences et des technologies. Les gens partagent facilement les connaissances ce qui conduit à des grands volumes de données textuelles accessibles sur l'internet. En raison du progrès rapide de la recherche scientifique, de plus en plus d'articles scientifiques sont publiés. Or, la grande majorité des données sont des données non-structurées et le volume des textes augmente à une vitesse telle qu'il est impossible de gérer les informations par les humains. Il est donc très important de fournir aux chercheurs des moyens automatisés et efficaces pour trouver les informations et leurs permettre de se tenir au courant de l'évolution scientifique au sein du domaine. L'objectif de fouille de textes est de surmonter ce problème [Cohen and Hunter, 2008]. Une de ses sous-tâches, l'extraction d'information sert à identifier et collecter les informations dans les textes, par exemple, pour extraire les entités biomédicales comme les noms de protéines, gènes, cellules etc sur les articles biomédicaux. La terminologie, c'est-à-dire une liste de termes d'un domaine, joue dans ce travail un rôle important pour améliorer la couverture d'un système d'extraction d'informations [Meystre et al., 2008, Cohen and Demner-Fushman, 2014]. Cependant, les ressources terminologiques sont difficiles et coûteuses à créer. La méthode traditionnelle de constitution d'une terminologie dépend des experts du domaine qui identifient les termes et les classifient. Mais le processus est coûteux à la fois en temps, en énergie et en argent. Beaucoup des travaux se consacrent donc à extraire automatiquement des informations de corpus pour créer les ressources terminologiques. En plus, une ressource devrait répondre à des besoins spécifique, surtout pour les corpus de spécialité. Le problème d'adaptation au corpus et de mise à jour des nouveaux termes limitent parfois leur utilisation [Bodenreider et al., 2002] [McCray et al., 2002]. Les méthodes automatisées pour aider à constituer et améliorer ces ressources sont donc cruciale. Pour réaliser cette phase de constitution de ressources adaptées, il est envisageable d'exploiter des méthodes d'extraction de termes

et d'analyse distributionnelle [Curran, 2004] de manière similaire à ce que a proposé [Grishman and He, 2014, Pazienza et al., 2005].

1.2 Objectif

L'objectif de notre travail consiste à combiner les méthodes d'extraction de termes et d'analyse distributionnelle afin de constituer une ressource adaptée au corpus. Le but est d'extraire sur le corpus les termes qui appartiennent à une même catégorie sémantique que les termes présentés dans la terminologie. Ce travail vise aussi à identifier le potentiel de la combinaison de l'extracteur de termes `YaTeA` qui fournit des termes candidats et l'outil d'analyse distributionnelle `word2vec` pour réaliser une tâche d'extraction d'information.

1.3 Définitions

Définition 1 (Terme)**.** *Un terme peut se définir comme un groupe de mots, en général un groupe nominal qui fait référence à une notion d'un domaine de spécialité.*

Définition 2 (Terme candidat)**.** *Un terme candidat est un groupe de mots extrait automatiquement par l'extracteur de termes* `YaTeA` *et peut se définir potentiel comme un terme.*

Définition 3 (Classe)**.** *Une classe présente un ensemble de termes qui appartiennent à une même catégorie sémantique.*

Définition 4 (Cluster)**.** *Un cluster est un ensemble de termes qui sont regroupés par un algorithme de classification non-supervisée.*

Définition 5 (Extraction d'information)**.** *L'extraction d'information consiste à extraire sur des textes une classe d'informations déterminées.*

Définition 6 (Terminologie)**.** *Une terminologie est un ensemble de termes d'un domaine de spécialité.*

1.4 Plan de lecture

Ce document est organisé comme suit : le chapitre 2 présente les travaux connexes. Dans le chapitre 3, nous décrivons le corpus et son prétraitement suivi d'une présentation de la tâche d'évaluation. Nous présentons, dans le chapitre 4, la méthode proposée pour l'analyse distributionnelle et le clustering. Nous présentons nos expériences et les résultats obtenus dans le chapitre 5 et nous concluons dans le chapitre 6.

CHAPITRE 2

ÉTAT DE L'ART

Sommaire

2.1 Fouille de textes

En 1988, Don Swanson a publié un article [Swanson, 1988] dans lequel il expose la découverte d'une relation entre la migraine et le magnésium en examinant manuellement les articles de MEDLINE. Cette découverte donne l'inspiration et encourage les chercheurs à proposer des méthodes automatiques pour fouiller dans les textes biomédicaux et détecter les informations implicites qui amènent les hypothèses scientifiques. Ce travail de détection a été beaucoup stimulé par les techniques de fouille de textes. Cette dernière ne cherche pas à comprendre le sens profond des textes mais à traiter efficacement certaines tâches précises, telles que la reconnaissance et la catégorisation d'entités biomédicales, l'extraction de terminologie. Le nombre de publications concernant le travail de la fouille de textes du domaine biomédical, surtout sur l'extraction d'information [Cohen and Hersh, 2005] et [Zweigenbaum et al., 2007]. Telles que la reconnaissance d'entités nommées à partir de la base de données MEDLINE, l'extraction de relations entre les entités du domaine et l'extension d'ontologies, etc. [Meystre et al., 2008] résument les recherches sur l'extraction d'informations à partir des dossiers médicaux électroniques.

Nous remarquons au-delà de ces études que les terminologies et les ontologies de spécialités sont souvent décrites comme des ressources dans les systèmes de fouille de textes. Ces ressources sont utilisées pour aider les tâches telles que l'identification des entités du domaine dans les textes de spécialité et l'extraction de relation.

2.2 Adaptation de ressources

Les ressources terminologiques sont beaucoup utilisées dans les applications du traitement automatique des langues. Dans l'extraction d'information, l'identification et la classification des termes sont les étapes essentielles. Les systèmes d'information utilisent une ressource terminologique. Il est donc nécessaire de constituer ou d'adapter la ressource terminologique. Certain travaux ont pensé à réutiliser les ressources potentielles. [McCray et al., 2002] ont analysé les propriétés lexicales de l'ontologie de gène (GO), celle-ci est développée dans le but de recreuser et d'annoter

les informations moléculaires et les produits des gènes. Ils ont cherché à évaluer si elle est appropriée en tant que ressource pour les applications de TAL. Les résultats indiquent que 79% des termes de la GO sont potentiellement utiles pour l'application de TAL. D'autres pensent à chercher les nouveaux termes dans les textes, [Bodenreider et al., 2002] ont extrait les syntagmes nominaux dans MEDLINE et le lexique d'UMLS, puis ils les ont comparés en fonction de leurs différences de modification pour découvrir automatiquement les nouveaux termes. 125,000 termes candidats sont identifiés, parmi les 1000 termes sélectionnés aléatoirement, 83% sont considérés pertinents.

Mais les ressources sont difficilement utilisées lorsqu'on rencontre les nouveaux termes ou quand on les applique sur les nouveaux corpus. Il est donc important de constituer ressources qui s'adaptent aux corpus. Différentes méthodes sont proposées ainsi pour constituer les ressources terminologiques à partir des corpus [Hersh et al., 1996, Pazienza et al., 2005, Cabré Castellví et al., 2001, Grishman and He, 2014]. Pour ce faire, certaines méthodes détectent les termes similaires aux termes existants. Il est possible d'identifier des mots similaires en utilisant l'hypothèse distributionnelle que les mots similaires apparaissent dans des contextes similaires. Cette approche consiste à utiliser les corpus pour examiner les contextes où chaque terme et en calculant ensuite la similitude entre les distributions de contexte des termes. [Grishman and He, 2014] ont construit un outil qui aide l'utilisateur à construire les classes d'entité et les relations spécifiques à un problème d'extraction d'information donné.

Un ensemble d'outils TAL permet de produire une liste de termes candidats. La liste de mots est ensuite utilisée comme un noyau dans une analyse distributionnelles pour récupérer les nouveaux termes ayant un certain degré de similarité syntaxique avec les noyaux. [Henriksson et al., 2011] ont utilisé la méthode de Random Indexing pour créer un espace vectoriel de mots et produire les codes ICD-10 recommandés. Il s'agit d'aider les médecins à assigner les codes aux documents des patients. L'idée de Random Indexing est de prendre des vecteurs de grande dimension et de les projeter de manière aléatoire dans un espace ayant relativement moins de dimensions [Sahlgren, 2005]. L'objectif des deux études n'est pas de détecter des nouveaux termes, mais l'idée principale est d'estimer la similarité distributionnelle. Dans une étude récente, [Muneeb et al., 2015] ont évalué la performance des deux modèles de mots, word2vec [Mikolov et al., 2013a] [Mikolov et al., 2013c] et le GloVe [Pennington et al., 2014] pour une tâche de capture des propriétés sémantiques des mots du domaine biomédical. [Miñarro-Giménez et al., 2015] ont appliqué le word2vec au corpus médical non structuré pour tester son potentiel d'identifier des relations cliniques.

2.3 Positionnement et Bilan

Basé sur les travaux que nous avons présenté, nous proposons une méthode d'extraction de termes et d'analyse distributionnelle. Afin de constituer une ressource adaptée, nous proposons de classer les termes extraits selon leurs similarité en appliquant une analyse distributionnelle de ces termes. L'analyse distributionnelle permet d'identifier les mots sémantiquement similaires en analysant leurs contextes partagés. Nous pouvons donc les utiliser pour regrouper les mots similaires.

Les travaux de [Muneeb et al., 2015] [Miñarro-Giménez et al., 2015] traitent le corpus en unité de mots. Mais les mots sont assez génériques. Un mot polysémique

provoque le risque d'avoir des ambiguïtés. Nous préférons donc prendre en compte les termes. Parce qu'ils représentent des concepts plus fins d'un domaine. De plus, l'application un extracteur de termes, nous permet d'extraire des termes présents dans le corpus.

CHAPITRE 3

MATÉRIEL ET ÉVALUATION

Sommaire

3.1 Corpus

Description de corpus

Le corpus que nous utilisons est le corpus GENIA [Kim et al., 2003]. Il a été créé pour le but de fournir un matériel de référence pour la fouille de textes du domaine biomédical. Il est constitué à partir des articles de la base de données bibliographiques MEDLINE retrouvés avec les requêtes de «human», «blood cells» et «transcription factors». 1 999 extraits d'articles ont ainsi été sélectionnés et annotés manuellement par les experts du domaine. Le corpus GENIA contient 436 967 mots, 18 546 phrases et 97 876 entités biomédicales annotées parmi lesquelles, 94 477 sont annotées à la fois par une classe sémantique et un lemme (Tableau 3.1). A partir de ces 94 477 annotations que nous extrairons une référence dont nous allons présenter plus tard.

Mots	436 967
Phrases	18 546
Entités biomédicales annotées	97 876
Entités biomédicales annotées en lemme et classe	94 477

TABLEAU 3.1 – Informations sur le corpus

Le corpus annoté est au format XML. Dans chaque fichier XML, les extraits d'articles et leurs titres sont numérotés et segmentés en phrases. Les phrases sont annotées avec les balises `<sentence>` et les entités biomédicales sont annotées par les balises `<cons>`. La balise `<cons>` possède deux attributs, «lex» et «sem», qui correspondent à la classe sémantique et au lemme. La classe sémantique d'une entité biomédicale est annotée comme la valeur de l'attribut «sem», et son lemme est annoté comme la valeur de l'attribut «lex». Nous prenons un extrait d'annotation d'une phrase dans le corpus GENIA comme exemple (Figure 3.1). Nous voyons bien que la

```
<sentence>
<cons lex="interferon"
    sem="G#protein_family_or_group">Interferons</cons> (<cons
    lex="IFN" sem="G#protein_family_or_group">IFNs</cons>) inhibit
    induction by <cons lex="IL-4" sem="G#protein_molecule">IL-4</cons>
    of multiple genes in <cons lex="human_monocyte"
    sem="G#cell_type">human monocytes</cons>.
</sentence>
```

FIGURE 3.1 – Exemple d'une phrase annoté du corpus GENIA

phrase *«Interferons (IFNs) inhibit induction by IL-4 of multiple genes in human monocytes.»* est annoté dans la balise <sentence>. Tous les termes qui ont été identifiés comme des entités biomédicales sont annotées avec la balise `<cons>`. Par exemple, le terme Interferons est annoté dans la première balise `<cons>` où son lemme «interferon» est la valeur de l'attribut «lex» et sa classe «protein_family_or_group" est annotée comme la valeur de l'attribut «sem».

Référence

Nous voulons évaluer non seulement les termes retrouvés par notre méthode mais aussi la classification des termes par rapport à leur catégorie sémantique. Pour faire cela, nous avons besoin d'une référence, c'est-à-dire une liste de termes avec leurs catégories sémantiques associées.

L'annotation sémantique du corpus GENIA est réalisée selon les 36 classes terminales de la taxinomie GENIA des 48 classes basées sur une classification d'entités de génétique [Kim et al., 2004]. À travers le corpus GENIA et grâce à ces annotations nous avons pu extraire chaque terme annoté dans la balise `<cons>`, son lemme et sa classe annotés selon la taxinomie GENIA. Cette référence est utilisée pour évaluer la performance de notre méthode. L'exemple ci-dessous est un extrait de la liste de terme-classe extraite sur le corpus (Tableau 3.2). Les termes dans la colonne de gauche sont associées à leur classe sémantique annotée à droite. Cette liste contient 35 456 termes correspondants aux 36 classes.

Terme	Classe
complementary_DNA	DNA_molecule
X-chromosome	DNA_molecule
T_cell_antigen_receptor	protein_molecule
FasL	protein_molecule
female_patient	multi_cell
healthy_control	multi_cell
wild-type_EBV	virus
wild-type_HIV	virus

TABLEAU 3.2 – Extrait de la référence

Regroupement des classes

Nous avons remarqué que dans l'annotation de corpus GENIA, il existe des termes qui sont annotés en différentes classes sémantiques. De plus, les classes sémantiques qui sont fournies un même terme se trouvent souvent sous une même classe supérieure dans la taxinomie GENIA (Figure 3.2). Par exemple, le terme «proto-oncogene» est annoté en classe «DNA_molecule» dans la première phrase (Figure 3.3), mais dans la deuxième phrase, il est annoté en classe «DNA_family_or_group». Ces deux classes sont toutes les deux sous-classes de la classe DNA. Nous avons donc regroupé ces classes dans leur classe hyperonyme. Nous avons aussi exclu la classe «other_name», car selon sa définition d'annotation GENIA, cette classe est sémantiquement vague. Le tableau 3.3 présente les 36 classes de la référence et le nombre de termes qu'elles contiennent avant regroupement.

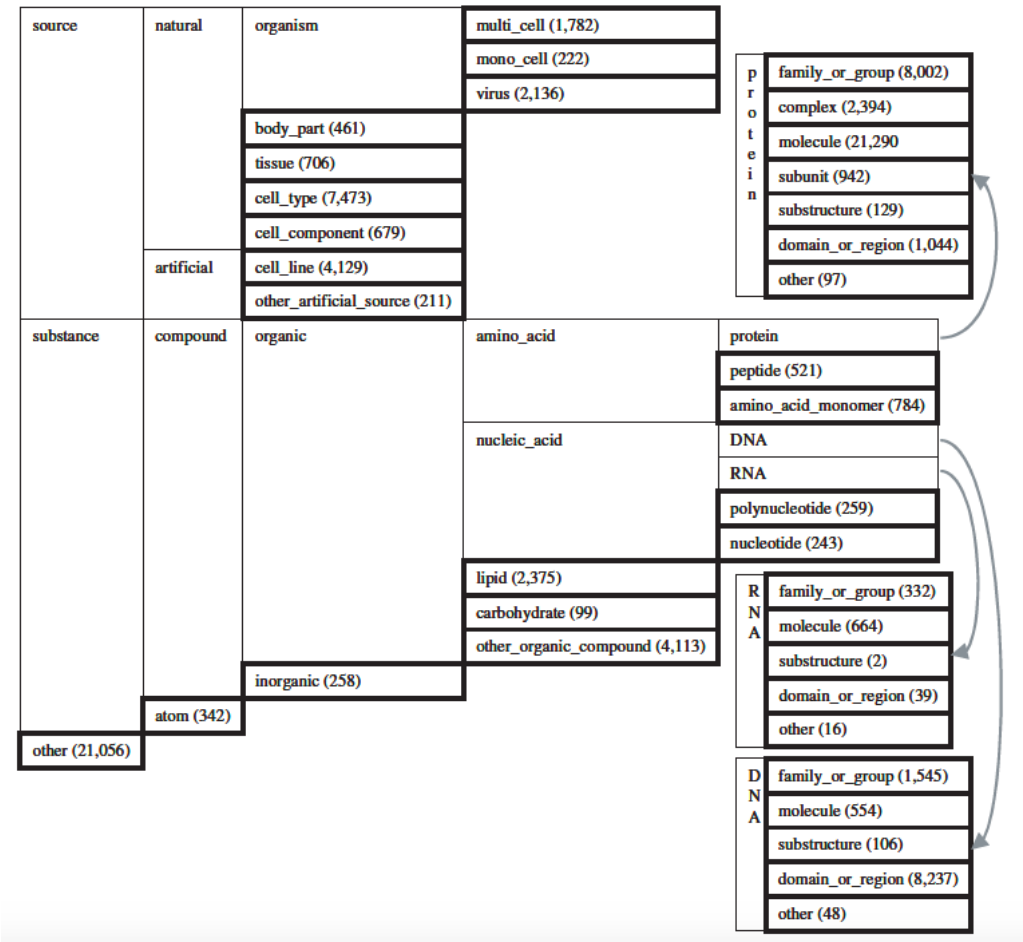

FIGURE 3.2 – Taxinomie GENIA [Kim et al., 2003]

Le résultat du regroupement est montré dans le tableau (Tableau 3.4). Les sous-classes dans la colonne de droite sont regroupées dans la classe dans la colonne de gauche. Après le regroupement des classes, la nouvelle référence contient 3 977 termes et 18 classes. Le tableau 3.5 présente les 18 classes de la nouvelle référence et le nombre de termes qu'elles contiennent.

```
<sentence>In conclusion , <cons lex="MDHM"
    sem="G#other_organic_compound">MDHM</cons>-stimulated induction of
    <cons lex="cytokine_mRNA" sem="G#RNA_family_or_group">cytokine
    mRNA</cons> expression was accompanied by different <cons
    lex="proto-oncogene" sem="G#DNA_molecule">proto-oncogene</cons>
    responses in <cons lex="PBMo" sem="G#cell_type">PBMo</cons> and
    <cons lex="THP-1_cell" sem="G#cell_line">THP-1 cells</cons>.
</sentence>

<sentence><cons lex="northern_blot_analysis"
    sem="G#other_name">Northern blot analysis</cons> revealed in these
    cells , an increase in the transcription of these two <cons
    lex="proto-oncogene"
    sem="G#DNA_family_or_group">proto-oncogenes</cons>, and this
    increase was amplified after treatment with <cons
    lex="phorbol_myristate_acetate"
    sem="G#other_organic_compound">phorbol myristate acetate</cons>.
</sentence>
```

FIGURE 3.3 – Exemple de phrases annotées du corpus GENIA

3.2 Prétraitement

Le prétraitement de notre corpus GENIA a été réalisé grâce à la plate-forme linguistique Ogmios [Hamon et al., 2007]. Cette plate-forme modulaire permet d'intégrer les ressources de spécialités et les outils de TAL spécialisés existants pour réaliser une série de traitements adaptés au corpus spécialisé : par exemple, la segmentation, la tokenisation, l'étiquetage morpho-syntaxique, la lemmatisation et l'extraction des termes. La configuration définie utilise l'étiqueteur morpho-syntaxique GENIA Tagger et l'extracteur de termes `YaTeA`. Le but de ces prétraitements est de préparer un corpus avec les informations nécessaires à l'analyse distributionnelle que nous réaliserons plus tard.

Étiquetage morpho-syntaxique et lemmatisation

Dans un premier temps, les mots sont étiquetés morpho-syntaxiquement et lemmatisés par l'étiqueteur morpho-syntaxique GENIA. Ce étiqueteur est créé spécialement pour le domaine biomédical [Tsuruoka et al., 2005]. À cause de la caractéristique spécifique des textes biomédicaux, les étiqueteurs morpho-syntaxiques pour les textes généraux fonctionnent moins bien sur ces textes. L'avantage de l'étiqueteur GENIA est qu'il est entraîné par à la fois les articles de langue générale du corpus Wall Street Journal et les articles biomédicaux du corpus PennBioIE [Mandel, 2006] et du corpus GENIA [Kim et al., 2003]. Il a donc une meilleure précision sur les corpus biomédicaux que les étiqueteurs morpho-syntaxiques généraux.

Extraction des termes candidats

L'extraction des termes candidats est réalisé par l'extracteur de termes `YaTeA` [Aubin and Hamon, 2006]. Il permet d'identifier les termes candidats, les groupes nominaux qui peuvent être des termes, par une analyse syntaxique superficielle sur le corpus étiqueté morpho-syntaxiquement et lemmatisé. Sa stratégie repose sur la

other_name	11330
protein_molecule	4527
DNA_domain_or_region	4256
protein_family_or_group	2950
cell_line	2254
cell_type	2004
other_organic_compound	1186
DNA_family_or_group	889
protein_complex	708
protein_domain_or_region	659
multi_cell	511
DNA_molecule	408
virus	386
tissue	384
protein_subunit	358
RNA_molecule	342
lipid	332
peptide	266
cell_component	232
body_part	194
polynucleotide	185
amino_acid_monomer	177
RNA_family_or_group	138
other_artificial_source	126
DNA_substructure	88
mono_cell	87
protein_substructure	86
protein_N/A	74
atom	68
nucleotide	66
inorganic	62
carbohydrate	47
DNA_N/A	34
RNA_domain_or_region	31
RNA_N/A	9
RNA_substructure	2

TABLEAU 3.3 – Nombre de termes des classes

détection de la frontière de termes, des règles d'analyse linguistique et sur une désambiguïsation endogène. Dans notre cas, il prend en entrée le corpus segmenté et étiqueté par l'étiqueteur morpho-syntaxique `GENIA` et fournit les termes candidats extraits. Ces termes seront utilisés pour la définition des mots cibles et des contextes dans l'analyse distributionnelle. `YaTeA` a extrait sur le corpus entier 77 573 termes. Le tableau 3.6 est une liste de termes candidats extraits par `YaTeA`.

organism	multi_cell mono_cell virus
artificial	cell_line other_artificial_source
protein	protein_family_or_group protein_complex protein_molecule protein_subunit protein_substructure protein_domain_or_region protein_N/A
DNA	DNA_family_or_group DNA_molecule DNA_substructure DNA_domain_or_region DNA_N/A
RNA	RNA_family_or_group RNA_molecule RNA_substructuresont RNA_domain_or_region RNA_N/A

TABLEAU 3.4 – Regroupement des classes

genital_tract_infection
OTF-2_expression_vector
oxidant-induced_NF-kappa_B_activation
danazol_therapy
large_clonal_T-cell_population
USF-related_transcription_factor
cytoplasmic_free_calcium
kidney
29-kDa_protein
untransfected_cell

TABLEAU 3.6 – Extrait de termes candidats

3.3 Méthode d'évaluation

Nous allons présenter dans ce chapitre, la méthode d'évaluation utilisée pour assurer la qualité de notre approche. En effet, la qualité de notre système de classification doit être mesurée en comparant les informations extraites par ce système avec une référence, c'est-à-dire les informations attendues. Plus les informations de ce système correspondent à celles de la référence, meilleur est ce système. L'évaluation du système permet de montrer la performance du modèle et aussi d'analyser les résultats. Nous pouvons découvrir les limites et les défauts du système et tenter de l'améliorer. Rappelons que l'objectif de notre travail consiste à classer les nouveaux

protein	9362
DNA	5675
artificial	2380
cell_type	2004
other_organic_compound	1186
organism	984
RNA	522
tissue	384
lipid	332
peptide	266
cell_component	232
body_part	194
polynucleotide	185
amino_acid_monomer	177
atom	68
nucleotide	66
inorganic	62
carbohydrate	47

TABLEAU 3.5 – Nombre de termes des classes

termes extraits automatiquement. Nous comparons donc les classes associées aux termes par notre système et les classes associées aux termes dans la référence, de manière à connaître la performance de notre système.

3.3.1 Division de corpus

En vue d'évaluer les termes extraits et leur classification, nous avons divisé de manière aléatoire le corpus GENIA en deux sous-corpus disjoints, un corpus de 1 200 fichiers (60% du corpus total) sert à entraîner le modèle et un corpus qui comprend 799 fichiers (40%) sont de corpus de test. En outre, pour la sélection optimale de paramètres, nous avons subdivisé le corpus d'entraînement en deux sous-ensembles, un ensemble d'apprentissage de 960 fichiers (80%) et un ensemble de développement de 240 (20%) fichiers. De la même manière, nous avons divisé la référence (Figure 3.4).

3.3.2 Mesures d'évaluation

La méthode d'évaluation choisie doit tenir compte des caractéristiques des résultats à évaluer. Notre objectif est d'évaluer la classification des termes extraits automatiquement. Les résultats sont les termes ordonnés et associés avec leur classe. Nous avons donc pris en compte non seulement la classification, mais aussi l'ordre des termes dans une classe.

La précision (Équation 3.1) est une mesure classique souvent utilisée avec le rappel (Équation 3.2). Pour un système d'extraction, le rappel est la proportion des éléments pertinents qui sont retrouvés (vrais positifs) par rapport au nombre total d'éléments pertinents (vrais positifs + faux négatifs), et la précision se réfère à la proportion d'éléments extraits correctement (vrai positifs), par rapport au nombre total d'éléments extraits par le système (vrai positif + faux positif).

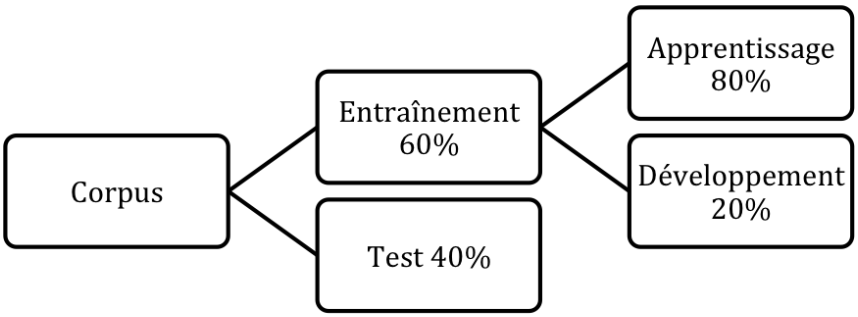

FIGURE 3.4 – Division de corpus

Vrai positif (VP) Les éléments pertinents retrouvés.

Faux positif (FP) Les éléments non-pertinents retrouvés.

Vrai négatif (VN) Les éléments pertinents non-retrouvés.

Faux négatif (FN) Les éléments non-pertinents non-retrouvés.

$$\text{Précision} = \frac{\text{nombre de éléments correctement retrouvés (VP)}}{\text{nombre de éléments retournés (VP+FP)}} \tag{3.1}$$

$$\text{Rappel} = \frac{\text{nombre de éléments correctement retrouvés (VP)}}{\text{nombre de éléments pertinents (VP+FN)}} \tag{3.2}$$

Mais la précision ne considère que le nombre d'éléments pertinents dans le résultat, sans tenir compte du regroupement et de l'ordonnancement entre des éléments. Or, nous voulons dans notre travail, non seulement évaluer la proportion de résultats pertinents qui sont retrouvés, mais aussi le classement de ces termes dans la liste des résultats. Plus les éléments pertinents sont rangés au début de la liste des résultats meilleur est le système. À ce propos, nous avons choisi la moyenne des précisions moyennes (MAP) (Équation 3.3). Cette méthode est beaucoup utilisée dans le domaine de la recherche d'information qui permet de calculer la précision moyenne de toutes les requêtes pour un modèle de Recherche d'Information. La mesure MAP est employée dans notre travail pour calculer la moyenne des précisions moyennes des termes extraits de toutes les catégories sémantiques.

$$MAP = \frac{\sum_{q=1}^{Q} AP(q)}{Q} \tag{3.3}$$

MAP Mean average precision d'un ensemble de classes est la moyenne de AP de chaque classe. q est une classe. Q est l'ensemble de classes.

Le Average Precision (AP) (Équation 3.4) calcule la précision de chaque position de la liste de résultats. Par exemple : si un élément non-pertinent se trouve dans une

position, la précision de cette position est 0. Et puis le AP calcule la moyenne des précisions de toutes les positions.

$$AP = \frac{\sum_{k=1}^{n} (P(k) \times rel(k))}{n} \tag{3.4}$$

n est le nombre d'éléments correctes.
$rel(k)$ est une fonction de l'indicateur égal à 1 si le terme au $k^{\text{ème}}$ rang est un terme pertinent, autrement égal à 0. Notons que la moyenne est sur tous les éléments pertinents et les éléments pertinents non récupérés obtiennent une précision de 0.
$P(k)$ est la précision en se limitant au $k^{\text{ème}}$ rang.

Quand à notre méthode, les termes sont classés en 18 classes. Dans ce cas, le MAP nous permet d'avoir un résultat unique. Les APs des classes seraient pondérés par la moyenne des APs de toutes les classes.

CHAPITRE 4

MÉTHODE

Sommaire

Nous allons décrire en détail dans ce chapitre, la méthode que nous avons mise en œuvre. Comme nous avons mentionné dans l'État de l'art du (chapitre 2), des travaux ont tenté de créer des ressources adaptées aux corpus et d'extraire les informations par l'analyse distributionnelle. Nous proposons à partir de ces travaux, une approche qui réalise une analyse distributionnelle des termes candidats extraits d'un corpus de spécialité. Il s'agit d'obtenir une représentation vectorielle des termes, puis de classer les termes grâce à un clustering sur leurs vecteurs.

4.1 Analyse distributionnelle

En sémantique distributionnelle, on suppose que les mots sémantiquement similaires apparaissent dans les mêmes contextes [Harris, 1954]. Cette hypothèse implique que le sens d'un mot peut être représenté par la distribution de ses contextes. L'analyse de la fréquence des contextes partagés permet de calculer la similarité des mots. Plus les mots partagent les mêmes contextes, plus ils sont sémantiquement proches. Nous pouvons ainsi capturer le sens de mots en représentant les contextes des mots dans un espace vectoriel de n-dimensions. Dans notre travail, nous nous intéressons aux termes extraits d'un corpus et qui appartiennent à la même classe sémantique. Pour ce faire, nous avons appliqué une analyse distributionnelle sur les termes de notre corpus, afin de les représenter dans un espace vectoriel. À partir de cette représentation vectorielle, nous pouvons estimer la similarité des termes en comparant la distance entre leurs vecteurs correspondants.

Limites de l'analyse distributionnelle

Lors de l'utilisation de l'analyse distributionnelle, deux limites peuvent être rencontrées : la dispersion des données dans la matrice des contextes et le problème de la très grande dimension des vecteurs [Perinet, 2015].

— **Dispersion des données** La matrice qui représente la distribution des mots est souvent très creuse c'est-à-dire la plupart des valeurs dans la matrice sont nulles.

— **Très grande dimension** Comme le nombre de dimensions correspond à la taille du vocabulaire des contextes, il peut être très grand et le calcul de la similarité des vecteurs est coûteux.

Pour résoudre ces problèmes, nous proposons dans notre étude, d'utiliser une approche qui prend en compte l'inclusion lexicale [Grabar and Zweigenbaum, 2004], afin de densifier la matrice des contextes et un modèle de Word Embedding pour éviter le problème des très grandes dimensions. Nous allons présenter ces méthodes dans les sections suivantes.

4.1.1 Présentation du Word Embedding

Word Embedding désigne l'ensemble des méthodes de modèles de langues, où les mots d'un vocabulaire sont représentés sous la forme de vecteurs de nombres réels dans un espace de faible dimension par rapport à la taille du vocabulaire. Autrement dit, il s'agit d'une représentation distributionnelle de mots apprise par des modèles de langue [Bengio et al., 2006, Collobert et al., 2011, Mikolov et al., 2013b]. Il permet de réduire les dimensions des très grands espaces vectoriels et de traiter le problème de la dispersion des données afin d'améliorer l'exactitude de la représentation sémantique des vecteurs. Le Word Embedding peut être calculée de différentes façons. Comme **Latent semantic analysis (LSA)**, le **Random Indexing** [Sahlgren, 2005] et une méthode plus récente, `word2vec`.

Présentation de `word2vec`

Pour obtenir une représentation vectorielle des termes, nous avons utilisé dans notre travail, une méthode efficace proposée par [Mikolov et al., 2013a] qui est implémentée dans un outil qui s'appelle `word2vec`. Cette méthode basée sur l'apprentissage de réseaux de neurones a tiré l'attention et a montré récemment une performance significative. Deux représentations des mots sont implémentées dans `word2vec`, qui sont le continuous bag-of-words (`CBOW`) et le `Skip-Gram` (Figure 4.1). Selon les auteurs [Mikolov et al., 2013a] :

— **CBOW** est pour objectif de prédire la probabilité d'un mot $w(t)$ sachant ses contextes $(w(t-2), w(t-1), w(t+1), w(t+2))$. Cette représentation de mots consomme moins de temps en entraînement que le skip-gram, et a une précision légèrement meilleure pour les mots fréquents.

— **Skip-Gram** contrairement aux CBOW, vise à prédire la probabilité des contextes d'un mot $(w(t-2), w(t-1), w(t+1), w(t+2))$ sachant ce mot $w(t)$. Elle fonctionne bien avec des corpus d'entraînement de petite taille et représente bien même les mots qui ont une fréquence faible.

Pour manipuler le `word2vec`, nous avons utilisé la librairie Gensim [1] [Řehůřek and Sojka, 2010]. C'est un outil de Text Mining écrit en Python, dans la-

1. Gensim Python library https ://radimrehurek.com/gensim/

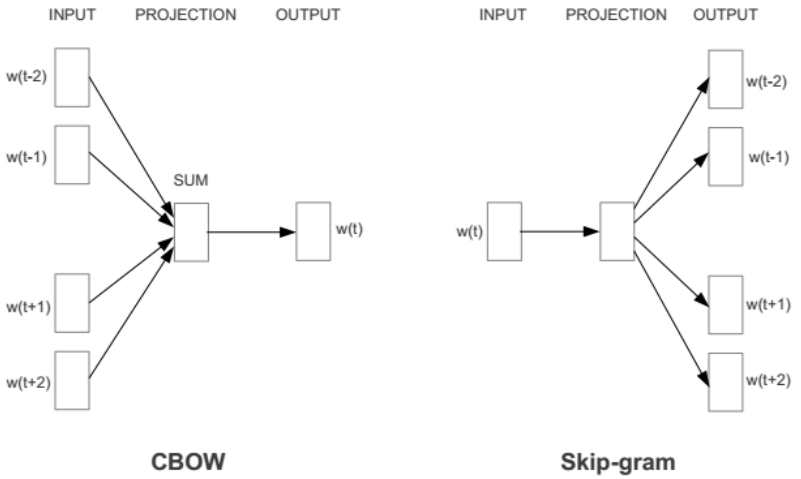

FIGURE 4.1 – Les algorithmes de word2vec

quelle l'algorithme `word2vec` est intégré.

4.1.2 Paramètres distributionnels

Dans les méthodes distributionnelles, la définition des paramètres est une étape de départ très importante. Avant d'appliquer `word2vec`, nous avons défini les mots cibles (les termes que nous allons classer) et les contextes des mots cibles (les mots voisins des mots cibles) et les paramètres tels que la taille de fenêtre graphique (le nombre de mots voisins avant et après le mot cible), la dimension d'espace vectoriel pour entraîner le modèle.

4.1.2.1 Définitions de contextes

Figement des termes candidats

Quand `word2vec` apprend à constituer la représentation de mots, il prend en compte l'espace comme séparateur. Il traite chaque token du corpus comme un mot. Ainsi, si un mot cible ou un contexte est un terme complexe, il sera divisé et représenté en mots isolés dans l'espace vectoriel. Nous nous intéressons non seulement aux mots cibles et aux contextes des termes simples, mais aussi ceux de termes complexes. Pour que ces termes complexes soient aussi représentés comme un seul élément, nous avons donc réalisé un figement de termes complexes. Pour cela, l'extracteur de termes `YaTeA` permet de donner une sortie dans laquelle les termes complexes sont figés. Nous voyons bien que dans l'extrait de la liste de termes candidats (Tableau 3.6), les termes complexes sont fusionnés par le tiret bas. Ces termes pourront ensuite être pris en compte comme un seul élément par `word2vec`.

Analyse syntaxique des termes candidats

L'approche d'inclusion lexicale est basée sur l'hypothèse que si un terme est lexicalement inclus dans un autre, il existe généralement une relation d'hyperonymie entre les deux termes [Grabar and Zweigenbaum, 2004]. Nous avons appliqué cette approche à travers l'analyse syntaxique de termes complexes four-

nie par YaTeA, afin d'augmenter le nombre de contextes. Par exemple, dans la phrase «Lymphocyte glucocorticoid receptor number in posttraumatic stress disorder.» 4.2, pour le termes «posttraumatic_stress_disorder», sa tête «disorder» et le constituant nominal «stress_disorder» sont inclus dans le terme maximal «posttraumatic_stress_disorder», idem pour le terme «Lymphocyte glucocorticoid_receptor_number» (Tableau 4.1). Nous réécrivons cette phrase en prenantes compte l'analyse syntaxique des termes complexe (Tableau 4.2). Dans notre exemple de tableau 4.2, la phrase est donc réécrite 11 fois. Cette application permet d'argumenter le nombre de contextes et de ne pas rester sur le figement de terme le plus grand. Nous laissons après à word2vec d'apprendre la représentation des mots avec ces contextes. Après le traitement d'augmentation des contextes, la taille du corpus est beaucoup plus grande (Tableau 4.3).

Terme le plus grand	**Lymphocyte_glucocorticoid_receptor_number**
Sous terme	Lymphocyte **glucocorticoid_receptor_numbe**
Sous terme	Lymphocyte glucocorticoid **receptor_number**
Sous terme	Lymphocyte glucocorticoid receptor number
Terme le plus grand	**posttraumatic_stress_disorder**
Sous terme	posttraumatic **stress_disorder**
Sous terme	posttraumatic stress disorder

TABLEAU 4.1 – Exemple d'analyse syntaxique des termes

Lymphocyte_glucocorticoid_receptor_number in posttraumatic stress disorder.
Lymphocyte_glucocorticoid_receptor_number in posttraumatic **stress_disorder**.
Lymphocyte_glucocorticoid_receptor_number in **posttraumatic_stress_disorder**.
Lymphocyte **glucocorticoid_receptor_number** in posttraumatic stress disorder.
Lymphocyte **glucocorticoid_receptor_number** in posttraumatic **stress_disorder**.
Lymphocyte **glucocorticoid_receptor_number** in **posttraumatic_stress_disorder**.
Lymphocyte glucocorticoid **receptor_number** in posttraumatic stress disorder.
Lymphocyte glucocorticoid **receptor_number** in posttraumatic **stress_disorder**.
Lymphocyte glucocorticoid **receptor_number** in **posttraumatic_stress_disorder**.
Lymphocyte glucocorticoid receptor number in posttraumatic stress disorder.
Lymphocyte glucocorticoid receptor number in posttraumatic **stress_disorder**.
Lymphocyte glucocorticoid receptor number in **posttraumatic_stress_disorder**.

TABLEAU 4.2 – Exemple d'augmentation de contextes

	Avant	Après
Mots	426 967	1 056 650 467
Phrases	18 564	13 577 401

TABLEAU 4.3 – Informations de corpus après l'augmentation des contextes

Normalisation de contexte

Dans les travaux existants en utilisant word2vec, plusieurs travaux ont réalisé des normalisations comme convertir en minuscules des mots, transformer les chiffres

en mots, enlever les ponctuations, etc. Cette normalisation permet une densification des contextes. Dans cette partie, nous présentons les quatre normalisations des textes que nous avons appliquées sur notre corpus : la normalisation des chiffres, la normalisation des ponctuations, la normalisation de casse et l'exclusion des stop-words.

Le corpus que nous utilisons est un corpus biomédical, il est donc nécessaire de tenir compte des caractéristiques orthographiques et morphologiques des termes biomédicaux et notamment leurs structures spécifiques. Ainsi, les termes biomédicaux contiennent souvent des chiffres et les symboles de ponctuation. Dans les corpus biomédicaux, nous pouvons également observer beaucoup d'unités de mesure, des formules chimiques, des abréviations et des nombres en chiffres romains, dans lesquels la distinction de majuscules et minuscules est nécessaire pour éviter les ambiguïtés. Pour ces raisons là, nous proposons de faire une normalisation plus ciblée.

Normalisation des ponctuations et de casse (NS) Nous avons normalisé les ponctuations qui sont accolées un terme mais qui ne font pas partie de ce terme. Les ponctuations éliminées sont les «.», «,», « ;», « :» à la fin d'une phrase, et les «()», «[]», «'», «+/-». Par exemple, pour le token «specific_autoantibody.», le point a été collé sur le terme «specific_autoantibody» est supprimé. Comme nous avons présenté, dans les textes biomédicaux, la normalisation de casse risque de générer des ambiguïtés. Dans notre corpus, les termes sont déjà lemmatisés, les termes biomédicaux gardent toujours leur casse. Mais les termes dans le début de la phrase sont toujours en majuscule. Comme nous avons décidé de garder la casse des termes, nous n'avons transformé que les mots qui ne sont pas les termes candidats en minuscule. Pour un terme candidat complexe, si son premier mot commence par une lettre majuscule et est suivis de minuscules, nous transférons ce premier mot en minuscule. Les mots normalisés sont les mots comme «Therefore», «A», mais les termes candidats comme «AP-1», «PMA» ne vont pas être normalisés.

Normalisation des chiffres (CH) Dans le travail de [Muneeb et al., 2015], les chiffres sont groupés dans les différents ensembles. Par exemple, les chiffres simples sont remplacés par «number1» et les chiffres doubles sont remplacés par «number2». En inspirant par ces opérations, nous avons normalisé les digits de différentes façons en fonction du type de nombres. Par exemple, tous les nombres à un chiffre sont remplacés par «$», les chiffres de pourcentage sont normalisés en «$%». Ces opérations sont effectuées sur les tokens isolés. Si les chiffres sont compris dans des termes complexes figés, nous ne les avons pas normalisés.

Exclusion des stop-words (GL/MO) Nous avons également utilisé un filtre par une liste de stop-words pour supprimer les mots qui ne portent pas ou portent moins de sens. Nous avons pour cela repris une grande liste de stop-words proposée par `Google`[2] et une liste de mots outils d'anglais[3] fournie par [Cook, 1989]. La liste de mots-outils contient 220 mots. Ce ne sont que les mots-outils, c'est-à-dire les mots qui ne portent pas forcément de sens et qui ont plutôt une fonction grammaticale, tels que les articles et les conjonctions. La grande liste de stop-words de `Google` a une taille de 828 mots. Elle contient non seulement les mots outils, mais aussi les adverbes (ex : usually, unfortu-

2. Stop-words, Google, https ://code.google.com/p/stop-words/
3. List of English structure (function) words, Vivian Cook, http ://homepage.ntlworld.com/vivian.c/Words/StructureWordsList.htm

nately), les noms de chiffre (ex : one, two, tree) etc. Ce sont les mots utilisés fréquemment dans la langue générale.

4.1.2.2 Paramètres de la représentation vectorielle

`word2vec` permet de faire la représentation de mots dans un espace vectorielle de d dimensions. Le nombre de dimensions d est typiquement entre 50 et 1000. Pour un vocabulaire de taille de 700000, il suffit de fixer le d à 300 comme dans [Mikolov et al., 2013a]. Nous avons dans le corpus total un vocabulaire de 79474 mots, donc nous avons fixé le d à 200. Dans notre étude, le corpus est issu d'un domaine de spécialité, il y a probablement les mots de spécialité qui ont des petites fréquences, nous avons donc choisi la représentation Skip-gram et nous avons fixé le nombre de dimensions à 200. Pour la taille de fenêtre graphique, nous avons fait référence au travail de [Perinet, 2015] dans lequel la taille de 21 était un paramètre adapté aux textes de spécialité et nous avons pris une taille approximative. Finalement, dans notre travail, `word2vec` a appris une représentation vectorielle des mots sur 200 dimensions, avec l'algorithme Skip-gram et une fenêtre de contextes de 20 mots.

4.2 Classification des termes

4.2.1 Regroupement avec k-means

Dans cette section, nous allons présenter la méthode que nous avons utilisée pour regrouper les vecteurs des termes appris par `word2vec`.

Présentation du k-means clustering

Dans le but de regrouper les vecteurs des termes similaires, nous proposons d'utiliser une méthode de classification non-supervisée, le k-means clustering. Cet algorithme est utilisé souvent en apprentissage non supervisé où l'on fait un regroupement de k partitions. L'objectif du k-means clustering est de regrouper les éléments dans un nombre fixé de (k) clusters, de façon à minimiser la distance d'un élément au centre de son cluster. Sa stratégie est de choisir d'abord aléatoirement k centres. Et puis, on regroupe les éléments en k groupes de manière à assigner aux k centres de départ les éléments les plus proches d'eux. Ensuite, on redéfinit la moyenne des k groupes comme les k centres. On refait le regroupement et on répète ces étapes jusqu'à les k centres ne changent plus ou le changement est très léger c'est-à-dire une stabilisation des partitions. Afin de manipuler le k-means clustering, nous avons utilisé Scikit-learn [4], un module Python intégrant les algorithmes classiques de l'apprentissage automatique dont le k-means clustering. La Figure 4.2 illustre un exemple de k-means clustering de deux clusters. Dans l'image `(b)`, deux centres de départ sont lancés aléatoirement. Puis, les élément sont regroupés dans le groupe le plus proche d'eux. Ensuite, les centres sont redéfinis par la moyenne des deux groupes `(d)`. Les étapes sont répétées `(e)`, jusqu'à la stabilisation des deux partitions `(f)`.

4. Scikit-learn Python library http ://scikit-learn.org/stable/

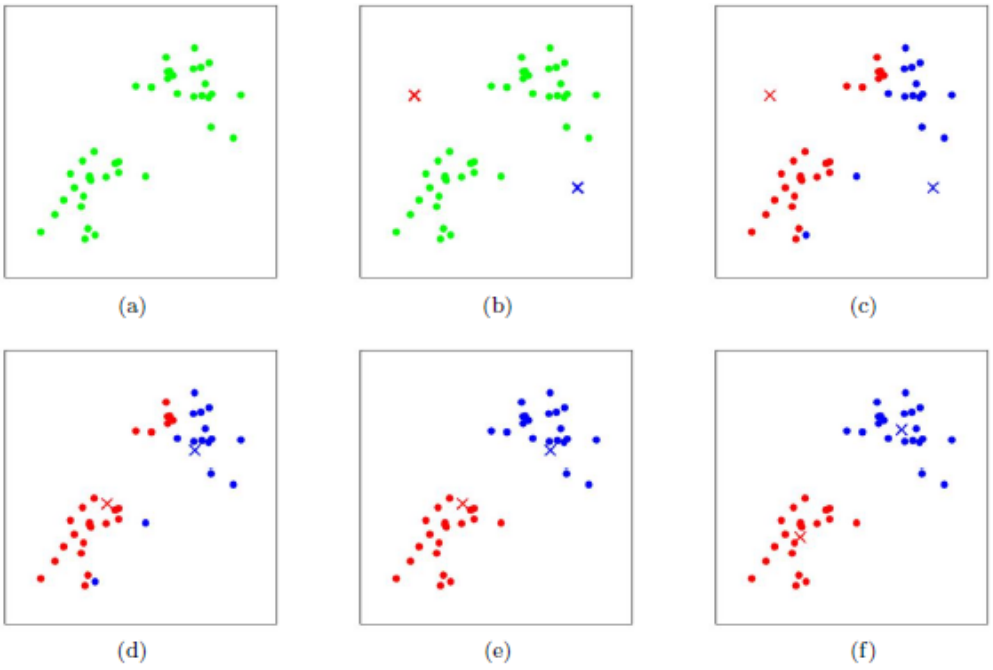

FIGURE 4.2 – Exemple de k-means clustering $k = 2$

Paramètre k

Pour le k-means clustering, le seul paramètre que nous devons fixer est le «k», c'est-à-dire le nombre de clusters. Cela présente un problème important : « Comment devrions-nous décider combien de clusters il faut créer ? » Le nombre de clusters (k) doit être fixé au préalable, mais nous avons aucune indication pour la définition de k. Pour trouver un nombre de k optimal, nous avons effectué plusieurs expériences en variant le nombre k. Nous avons fait ces expériences sur le corpus d'apprentissage et nous avons évalué les résultats sur le corpus de développement. Une fois le paramètre k défini, nous l'utilisons sur le corpus entier.

Sélection des vecteurs

Une fois le modèle appris par `word2vec`, nous avons une matrice de mots et leurs représentations vectorielles dans un espac à 200 dimensions. Puis, nous avons extrait les vecteurs des termes cibles, c'est-à-dire les termes candidats extraits par `YaTeA`. Nous avons ensuite effectué un regroupement de k-moyennes sur ces termes cibles. Les termes dont les vecteurs sont proches sont regroupés.

4.2.2 Classification des termes

Grâce à la représentation vectorielle des termes et le k-means clustering, les termes similaires se regroupent dans les k clusters. Mais à ce niveau de notre approche, nous ne savons pas à quelle classe sémantique correspondent ces groupes de termes. Dans cette section, nous allons décrire la méthode de détection de classe sémantique des termes. La méthode que nous allons présenter dans cette partie consiste à la fois à trouver le paramètre k et détecter et classer les termes.

4.2.2.1 Vote majoritaire

Rappelons que nous avons créé une référence de termes classés dans laquelle les termes sont associés à leur classe sémantique annotée. Nous avons donc à l'aide de ces termes-là détecté la classe sémantique des termes regroupés par le k-means clustering. Pour ce faire, nous faisons une hypothèse de vote majoritaire. Nous supposons que si la plupart des termes dans un groupe appartiennent à une classe, tous les termes de ce cluster devraient être dans la même classe.

Nous avons d'abord utilisé `word2vec` pour apprendre la représentation vectorielle des mots sur le corpus entier et nous avons ensuite regroupé les termes cibles. Nous avons regardé dans chaque groupe de termes s'il y a des termes appartenant à la référence du corpus d'entraînement. Et puis, nous avons fait un vote majoritaire en comptant parmi ces termes du corpus d'entraînement, la classe où la plupart de ces termes appartiennent. Par exemple (Tableau 4.4), dans ce groupe de 10 termes, il y a 8 termes du corpus d'entraînement et parmi ces 8 termes, il y a 7 termes appartenant à la classe «protein» et 1 terme appartenant de la classe «DNA». Alors tous ces 10 termes seront classés dans la classe «protein». Les termes appartenant à un groupe ne contenant aucun terme d'entraînement seraient abandonnés.

Classe	Terme
protein	alpha_B2
protein	PEBP2_alpha_A1
protein	alpha_A-gene_product
protein	alpha_B2_protein
protein	alpha_B-encoded_isomer
protein	A1
protein	alpha_A1
DNA	mouse_GM-CSF_promoter
-	alpha_B1
-	PEBP2_site

TABLEAU 4.4 – Exemple d'un groupe de termes

Poids de vote Nous constatons que dans la référence, les termes des classes sont déséquilibrés. Si nous prenons le poids d'un vote à 1, les classes qui ont moins de termes risquent d'avoir zéro dans le vote. Pour résoudre ce problème de déséquilibrage de nombres de termes dans les classes, nous avons fixé le poids d'une classe à l'inverse du nombre de termes qu'elle contient. Si une classe contient 99 termes, son poids est donc à 0.01010101 $(1/99)$.

4.2.2.2 Mise en ordre des termes

Après que nous avons classé les termes dans une classe, il nous faut aussi l'ordonnancement des termes. Nous préférons que les termes plus pertinents se trouvent au début de la classe. À ce propos, nous avons calculé d'abord la moyenne des vecteurs des termes d'un groupe et nous avons pris la moyenne comme le centre de ce groupe de vecteurs. Et puis, nous avons ordonné ces termes selon le cosinus de l'angle entre un terme et le centre de ce groupe. La mesure cosinus nous permet de calculer

la similarité entre deux vecteurs à n dimensions. La valeur $\cos\,\theta$ est comprise dans l'intervalle [0,1]. Plus la valeur est proche de 1, plus les vecteurs sont similaires.

$$\cos\theta = \frac{A \cdot B}{\|A\| \cdot \|B\|} \tag{4.1}$$

A et B sont les vecteurs.

Nous avons donc ordonné les termes par rapport à la similarité entre un terme et le centre de son groupe. Nous avons placé ces termes ordonnés dans la liste de termes de la classe à la quelle ils appartiennent. Prenons l'exemple du groupe de termes (Tableau 4.4), après le classement de similarité, les termes sont ordonnés comme dans (Tableau 4.5). Les termes de ce groupe sont bien ordonnés par leurs similarités du centre du groupe. Les termes qui sont les plus caractéristiques d'une classe sémantique seront placés plus en avance dans la liste de cette classe.

Classement	Terme	Similarité de centre
1	alpha_B2	0.921869340019
2	PEBP2_site	0.907329190049
3	alpha_A1	0.905462761638
4	alpha_B1	0.895985788505
5	alpha_B2_protein	0.882353176702
6	mouse_GM-CSF_promoter	0.863951263317
7	alpha_B-encoded_isomer	0.852014312837
8	alpha_A-gene_product	0.85005669816
9	PEBP2_alpha_A1	0.846504552976
10	A1	0.646963955369

TABLEAU 4.5 – Exemple d'un groupe de termes ordonnés

4.3 Sélection de paramètres

Ainsi pour sélectionner les meilleurs paramètres, nous avons d'abord effectué les expériences sur le corpus d'apprentissage et évalué les résultats avec le corpus de développement. Rappelons qu'il nous reste un travail d'optimisation du nombre de groupes (k). Dès que nous avons trouvé un meilleur paramètre de k, nous avons réalisé l'opération sur le corpus entier. Pour déterminer le nombre de k, la proportion de termes perdus, c'est-à-dire les termes de référence de développement abandonnés à cause d'absence de termes d'apprentissage dans le groupe sont également pris en compte. Nous allons décrire cette application dans le chapitre 5. La figure 4.3 illustre la processus de la méthode d'optimisation des paramètres. La figure 4.4 illustre le processus de notre méthode.

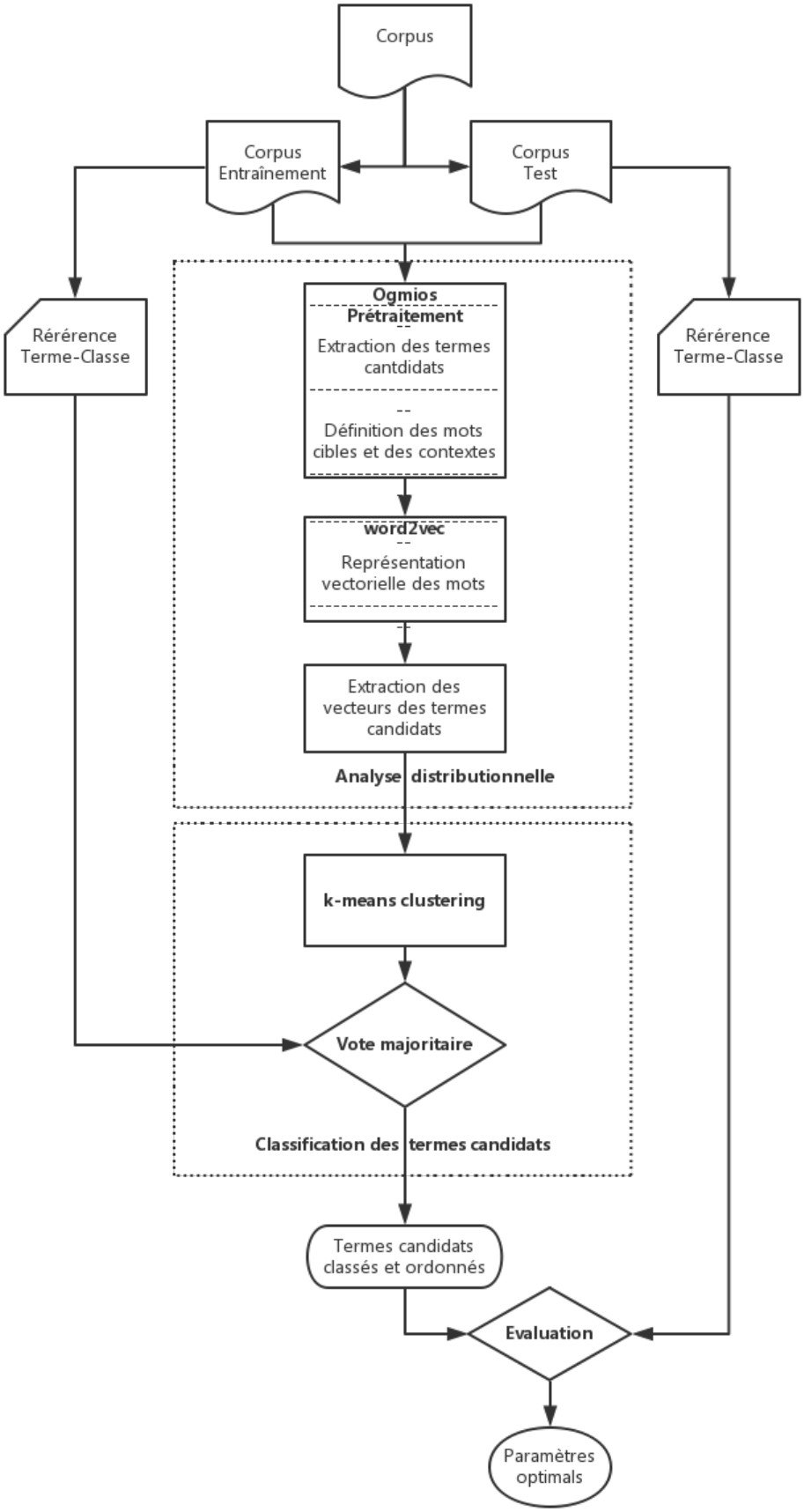

FIGURE 4.3 – Schéma de la méthode d'optimisation des paramètres

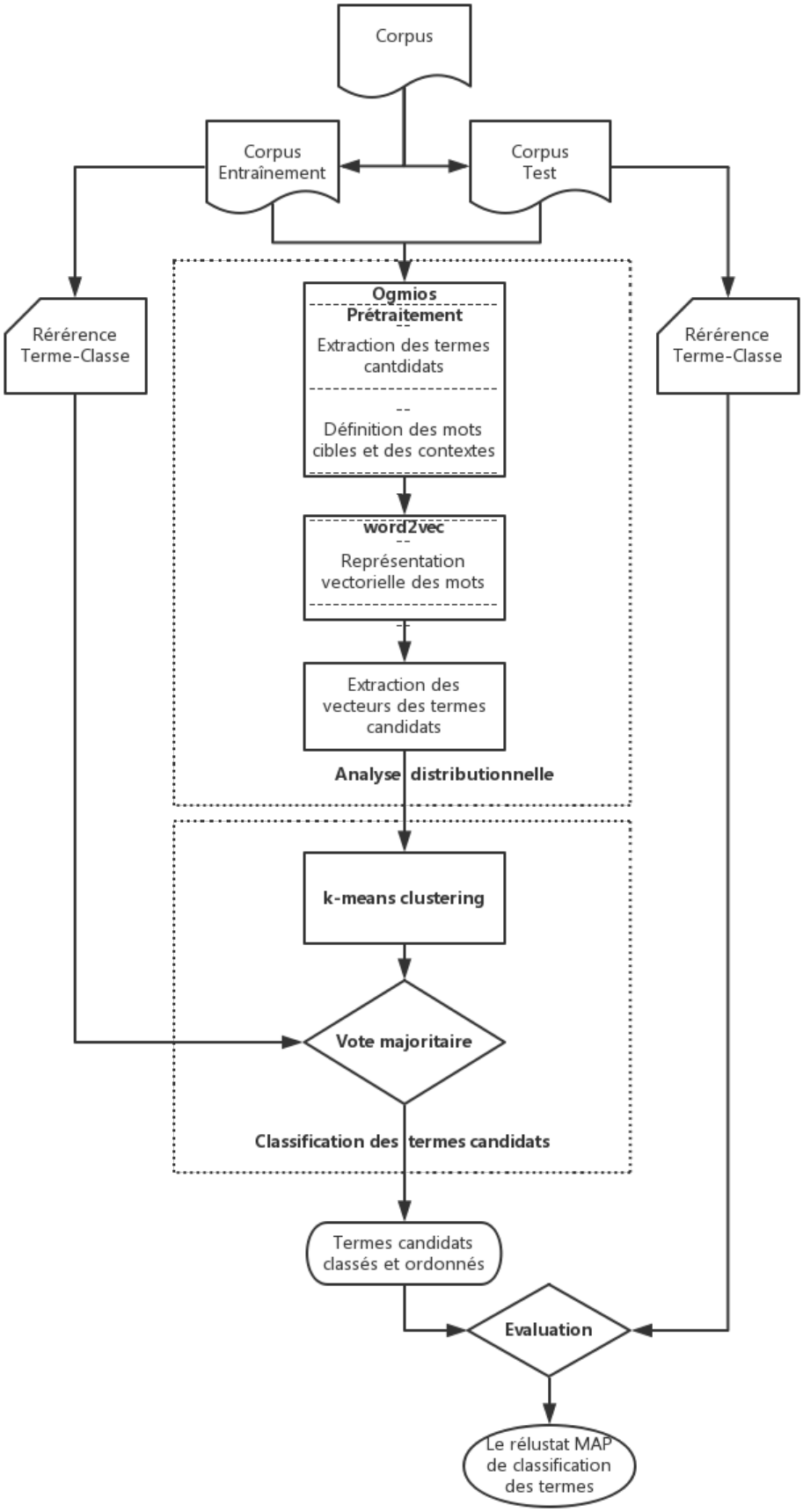

FIGURE 4.4 – Schéma de la méthode de classification des termes

CHAPITRE 5

EXPÉRIENCES ET RÉSULTATS

Sommaire

Nous présentons dans ce chapitre les expériences que nous avons effectuées et l'analyse des résultats obtenus. Rappelons que notre objectif est de classifier les termes candidats extraits. Afin de parvenir à un tel objectif, nous tentons de trouver la meilleure configuration du modèle pour l'analyse distributionnelle et pour la classification de k-means clustering. C'est-à-dire les différents tests que nous avons effectués nous permettent d'arriver à la configuration optimale. Cette configuration peut fournir les meilleurs résultats possibles sur le corpus de développement.

5.1 Expériences

Nous avons réalisé d'abord un ensemble d'expériences afin d'optimiser la méthode de normalisation de contextes. Dans le chapitre 4, nous avons proposé quatre méthodes de normalisation de texte. Pour choisir une méthode parmi ces quatre, nous les avons toutes appliquées et comparé leurs résultats. Les cinq expériences que nous avons réalisées sont :

NS normalise juste les ponctuations qui sont collées sur les termes.

NS+CH est une normalisation de NS et une normalisation des chiffres.

NS+MO est une normalisation de NS et une exclusion d'une liste de mots-outils proposée par Vivian Cook [Cook, 1989].

NS+CH+MO est une normalisation combinée de NS+CH+MO.

NS+CH+GL est une normalisation combinée de NS+CH plus une exclusion de la stop-liste proposée par `Google`.

Rappelons que l'algorithme k-means que nous utilisons pour la classification des termes candidats prend comme paramètre le nombre de classes k. Pour fixer le k, il est nécessaire de laisser ce paramètre libre, et de trouver le meilleur k possible. Afin d'automatiser la recherche du meilleur paramètre k, nous avons appliqué le k-means clustering plusieurs fois en incrémentant k à chaque fois. Nous avons fait autant d'expériences pour toutes les cinq méthodes que nous avons mentionnées précédemment.

Nous avons d'abord varié le k entre 500 et 1500 avec un intervalle de 100. Puis, le k-means clustering est appliqué en faisant varier le k entre 2000 et 5000 avec un intervalle de 1000.

Les expériences sont réalisées sur le corpus d'entraînement, et la référence utilisée pour évaluer la performance est la référence de développement. Une fois que nous avons fixé les paramètres qui conviennent à notre travail, nous les utilisons pour réaliser une expérience sur le corpus de test. La référence d'évaluation utilisée sera la référence de test.

5.2 Résultats

Nous avons évalué les expériences que nous avons présentées dans la section précédente en utilisant la mesure de MAP pour évaluer les résultats. Dans cette section nous présentons les résultats obtenus et nous les analysons.

5.2.1 Comparaison des méthode de normalisation de contextes

Le tableau 5.1 présente les résultats des expériences que nous avons réalisées. La première colonne à gauche indique le nombre de clusters k. Les valeurs dans les cellules correspondent aux résultats de MAP. Nous avons remarqué que la performance augmente avec l'augmentation du nombre de clusters k. Plus la taille de clusters, c'est-à-dire le nombre de termes d'un cluster est petite, meilleure est la performance.

k	NS	NS+CH	NS+MO	NS+CH+MO	NS+CH+GL
500	0,0015	0,0017	0,0022	0,0018	0,0016
600	0,0017	0,0017	0,0017	0,0022	0,0022
700	0,0019	0,0021	0,0024	0,0023	0,0024
800	0,0019	0,002	0,0022	0,0026	0,0027
900	0,0023	0,0023	0,0025	0,0026	0,0021
1000	0,0024	0,0024	0,0028	0,0025	0,0023
1100	0,0026	0,003	0,0027	0,003	0,003
1200	0,0028	0,0027	0,0029	0,0029	0,0032
1300	0,0029	0,0029	0,0032	0,003	0,0033
1400	0,0032	0,0031	0,0031	0,003	0,0033
1500	0,003	0,003	0,0034	0,0034	0,003
2000	0,0037	0,0036	0,0038	0,0037	0,0035
3000	0,0044	0,0045	0,0042	0,0047	0,0047
4000	0,0054	0,0061	0,005	0,0055	0,005
5000	0,0061	**0,0064**	0,0059	0,006	0,0061

TABLEAU 5.1 – Les MAP obtenues par les expériences des cinq méthodes de normalisation en variant le k

Les figures 5.2 et 5.1 illustrent l'évolution de MAP obtenue par les cinq méthodes appliquées en faisant varier le nombre de clusters k. En observant les résultats entre $k = 500$ et $k = 5000$, nous remarquons que les courbes des cinq méthodes de normalisation de contextes fluctuent et s'entrecroisent entre elles. Il est donc difficile de faire le choix entre les cinq méthodes.

Comme les courbes s'entrecroisent beaucoup, il reste difficile de les comparer. Nous avons donc fait un test de la significativité entre les résultats des 5 méthodes.

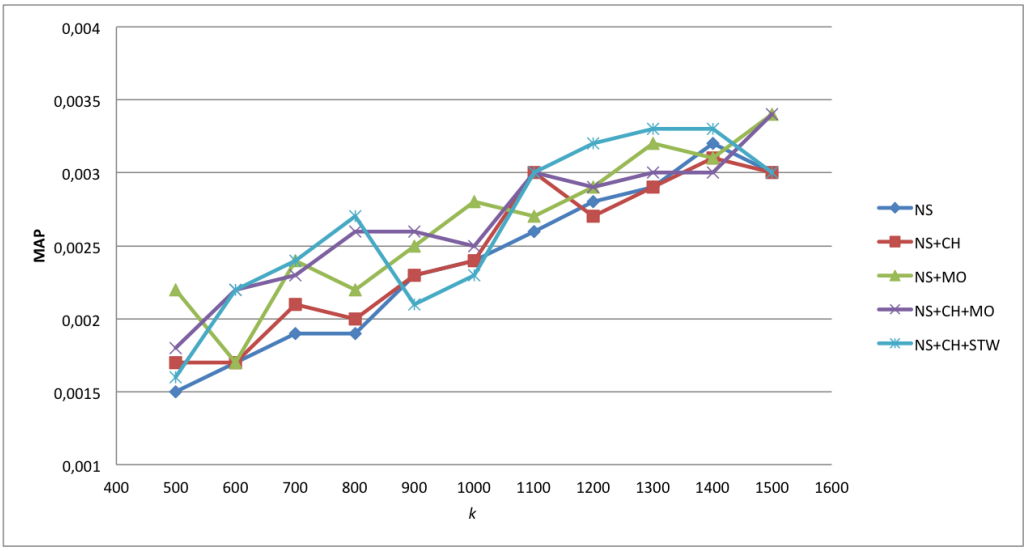

FIGURE 5.1 – Résultats de MAP en incrémentant k

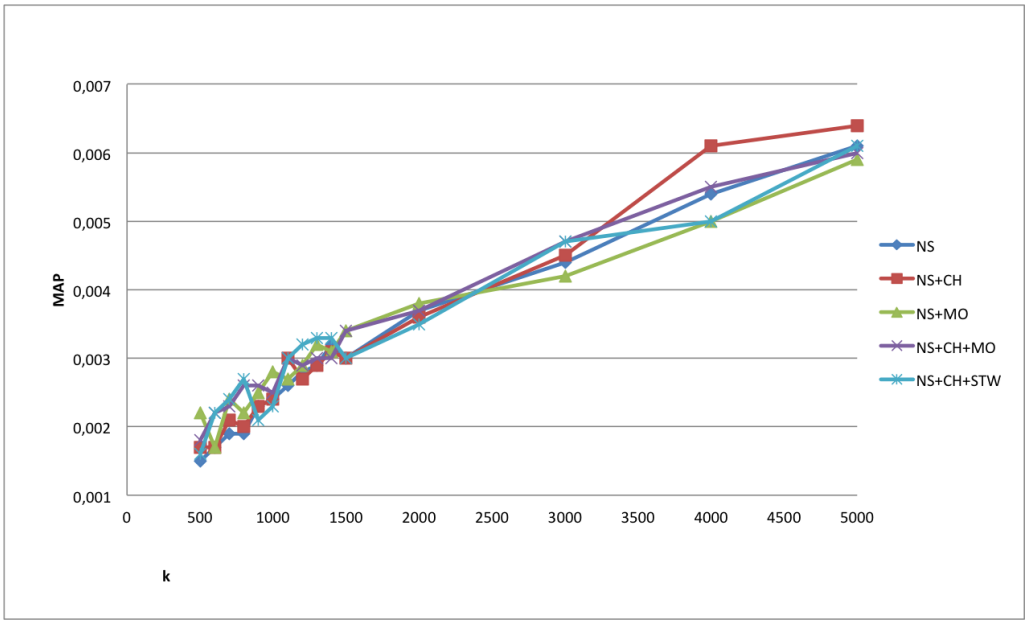

FIGURE 5.2 – Résultats de MAP en incrémentant k

Pour ce faire, nous avons réalisé un test de la significativité, le test de Student. Ce test permet de comparer deux groupes d'échantillons indépendants. Le test de Student donne en résultat deux valeurs, la t-score et la p-value. Ce qu'il faut regarder avant tout, c'est la p-value. Elle correspond à la probabilité que les deux échantillons considérées ne diffèrent pas significativement. Donc plus la p-value est petite, plus il est incontestable que les deux échantillons diffèrent. Par convention, on se fixe souvent le seuil de 0,05. Donc si la p-value est inférieure à 0,05, on peut dire que les deux échantillons ont une différence statistiquement significative.

Nous avons comparé les performances des cinq méthodes par paire et calculée la significativité de la différence entre elles. Le t-score présente la différence des deux groupes d'échantillons. Quand il n'y pas de différence entre le premier groupe et le

deuxième, le t-score est 0. Nous constatons dans les résultats obtenus (Tableau 5.2) que les valeurs de t-score sont toutes négatives, cela montre que la deuxième méthode a une meilleure performance que la première. Le CH est meilleure que NS. Mais, nous remarquons que sauf la p-value de **NS vs. NS+CH+MO**, toutes les p-value obtenues sont supérieure à 0,05, c'est-à-dire, la performance entre ces paires de méthodes n'est pas significatives. Cela montre que le traitement de la normalisation des chiffres, l'exclusion des mots outils et une grande liste améliorent légèrement la performance. La p-value obtenue de **NS vs. NS+CH+MO** égale à 0,003, elle est inférieure de 0,05, ce qui est suffisamment petit pour dire que la différence entre les résultats des deux méthodes est statistiquement significative. La performance de la normalisation combinée par la normalisation des chiffres et l'exclusion de mots-outils (NS+CH+MO) est meilleure significativement que la normalisation de ponctuations (NS). Dans ce cas, nous avons pris la méthode de NS+CH+MO comme la méthode optimale pour l'étape de la normalisation dans notre travail.

	t-score	p-value
NS vs. NS+CH	-1.996	0.066
MO vs. NS+CH+MO	-1.039	0.316
NS vs. NS+MO	-1.911	0.077
CH vs. NS+CH+MO	-1.410	0.180
CH+GL vs. NS+CH+MO	-0.759	0.461
NS vs. NS+CH+MO	-3.651	**0.003**
NS vs. NS+CH+GL	-2.031	0.062

TABLEAU 5.2 – Résultats des tests de significativité des performances des 5 méthodes de normalisation

5.2.2 Choix du nombre de clusters k

Nous avons choisi la méthode de la combinaison de la normalisation simple, la normalisation des chiffres et l'exclusion des mots-outils (NS+CH+MO), nous prenons les résultats de cette méthode pour faire le choix de k. Dans la figure 5.3, les MAP obtenus en fonction de nombre de clusters, nous observons que la courbe fluctue et qu'elle a tendance à augmenter avec l'incrément de k. Dans ce cas, nous nous sommes tournés vers les clusters perdus pendant le vote. Rappelons que nous avons fait un vote majoritaire pour définir la classe d'un cluster. Nous regardons d'abord si un cluster contient les termes de la référence d'apprentissage. Si la plupart des termes de la référence d'apprentissage dans ce cluster appartiennent à une même classe, nous classons tous les termes de ce cluster dans cette classe. S'il n'y a aucun terme de la référence d'apprentissage, ce cluster de termes est mis de coté, c'est-à-dire que les termes de ce cluster ne seront pas classés. Mais parmi ces termes, il peut exister des termes de la référence de développement. Ainsi le nombre de termes abandonnés est aussi un critère pour faire le choix de k. Nous avons donc examiné le nombre des termes de la référence de développement perdus à cause de l'absence des termes d'apprentissage.

La figure 5.4 illustre l'évolution du nombre de termes de la référence de développement perdus à cause d'absence des termes d'apprentissage en incrémentant de k. Nous remarquons qu'à partir de 1100 clusters, nous perdons plus de 5% de termes,

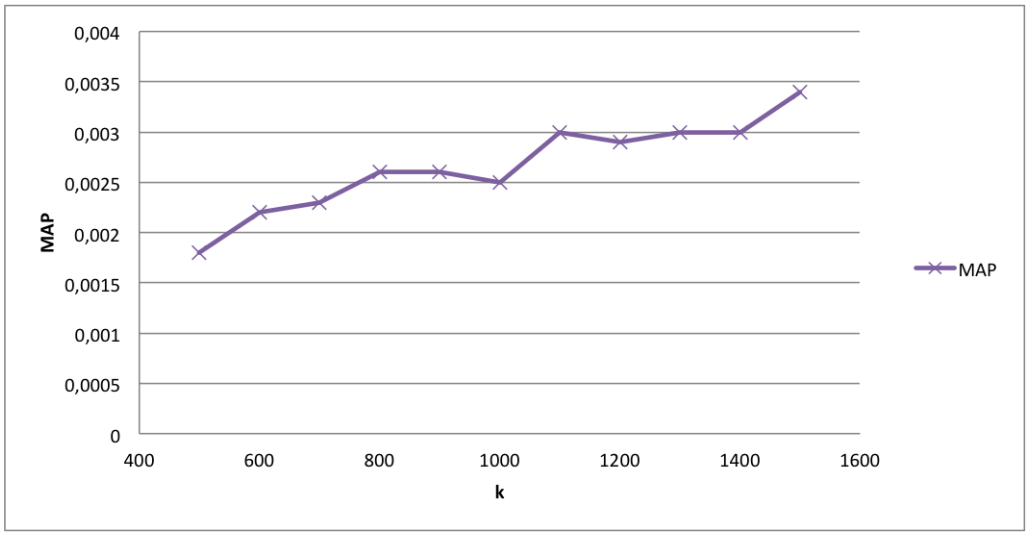

FIGURE 5.3 – MAP obtenus en fonction de nombre de clusters avec la méthode NS+CH+MO

ce qui nous semble assez important. Nous avons décidé de fixer à 1000 le nombre de cluster.

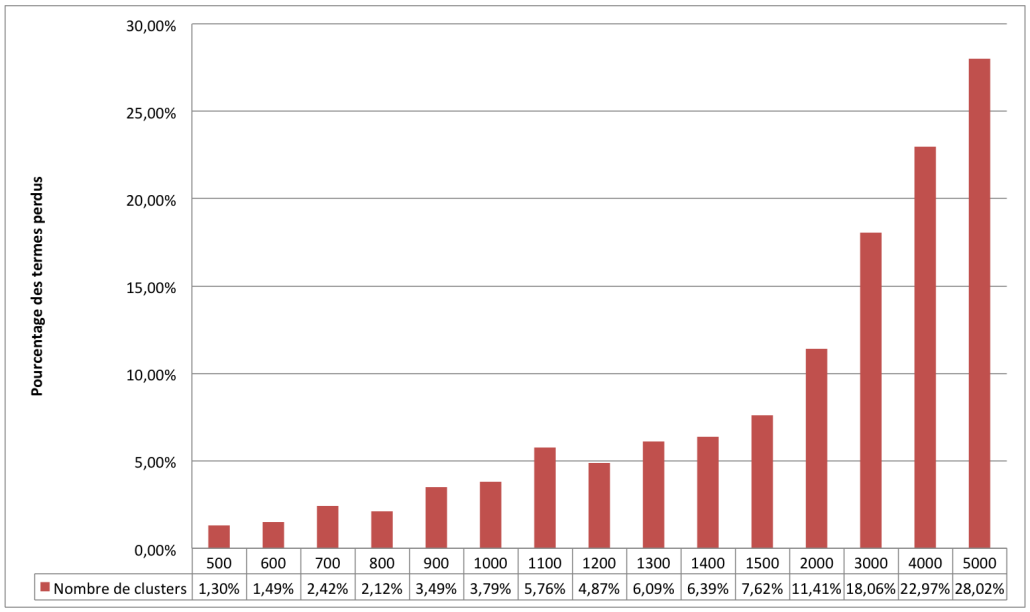

FIGURE 5.4 – Pourcentages des termes perdus

5.2.3 Performance de la méthode

Nous prenons les paramètres que nous avons choisis à travers les expériences effectuées sur le corpus d'entraînement pour évaluer la performance de notre méthode. Les paramètres choisis sont la normalisation combinée de la normalisation des chiffres et l'exclusion des mots-outils (NS+CH+MO) et le nombre de k égal à 1000. Nous avons appliqué notre méthode avec ces paramètres sur le corpus entier. La figure 5.5 illustre les résultats de AP de chaque classe (La précision moyenne de toute

les positions d'une liste de termes d'une classe) et le MAP (la moyenne des précisions moyennes de toutes les classes) de l'ensemble des classes obtenus par notre méthode.

Soulignons tout d'abord que pour toutes les classes, nous obtenons un résultat valable, aucune classe n'a obtenu un résultat nul. Au vu des AP des classes, nous constatons que les classes qui ont plus de termes dans la référence d'entraînement comme les classes de `DNA`, `protein`, `other_organic_compound` (Tableau 3.5) ont les meilleures AP. Alors que les classes qui ont moins de termes dans la référence d'entraînement comme `carbohydrate`, `atom` ont une petite valeur d'AP. Par contre, nous remarquons que la classe `body_part` n'a pas beaucoup de termes d'entraînement, mais elle a eu une AP relativement bonne. Nous supposons que la classe de `body_part` est peut-être sémantiquement loin d'autres classes.

Nous avons obtenu une MAP de $0,0027$. Nous supposons qu'il y a plusieurs raisons qui expliquent ce résultat faible. D'abord, nous ne prenons en compte que les termes candidats extraits par l'extracteur de termes. Les autres mots sont donc abandonnés. Nous perdons probablement les termes qui ne sont pas extraits. En outre, dans l'étape de vote majoritaire, nombre de termes sont abandonnés à cause de l'absence de termes d'apprentissage. En plus, sachant que les listes de termes candidats classés et ordonnés de chaque classe est beaucoup plus grandes que les listes de références. Les termes pertinents peuvent être classés en fin de liste.

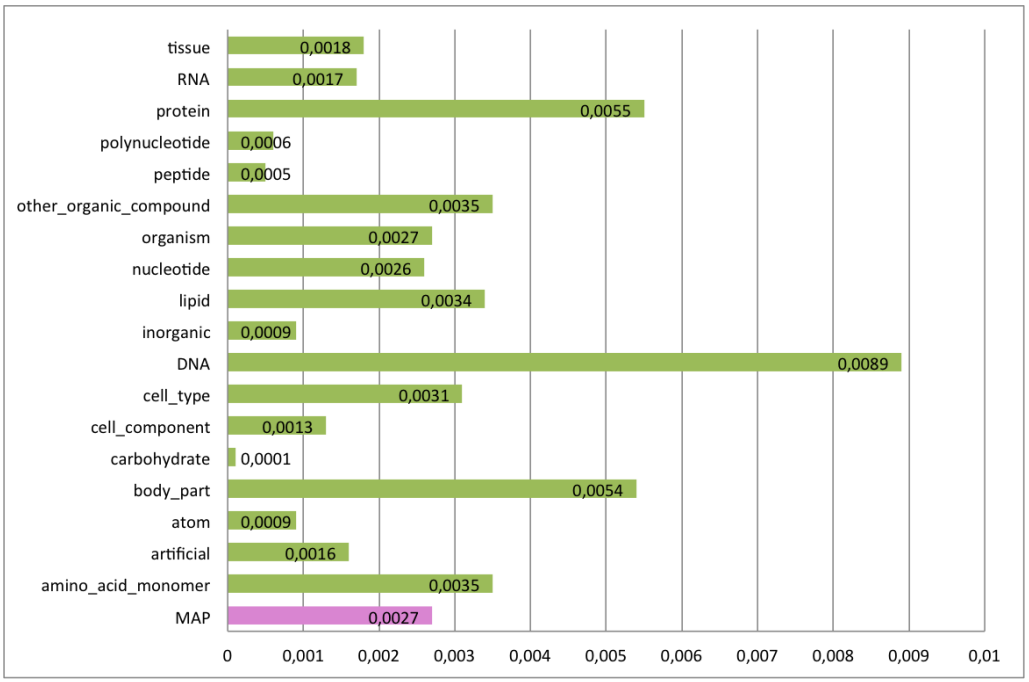

FIGURE 5.5 – AP des classes et le MAP d'ensemble des classes

CHAPITRE 6

CONCLUSION ET PERSPECTIVES

Sommaire

6.1 Conclusion

Nous avons présenté dans ce mémoire une méthode combinant l'extraction de termes et l'analyse distributionnelle pour classifier les termes d'un corpus. En vue de réaliser cet objectif nous avons d'abord extrait les termes candidats par l'extracteur `YaTeA` sur le corpus GENIA. Les termes candidats extraits sont ensuite utilisés pour définir les mots cibles et les contextes d'une analyse distributionnelle. Nous avons effectué cette analyse distributionnelle par `word2vec`. Les termes sont représentés comme les vecteurs dans un espace vectorielle de 200 dimensions. Cela permet de regrouper les termes similaires, c'est-à-dire les termes qui partagent les mêmes contextes. Le k-means clustering a été ensuite utilisé pour regrouper les termes candidats selon leurs similarités. Nous avons classé finalement les clusters obtenus dans les 18 classes GENIA.

Nous avons effectué un ensemble d'expériences afin d'obtenir les paramètres optimaux pour la définition des contextes d'analyse distributionnelle et pour le nombre de clusters de k-means clustering. D'après les résultats obtenus, la combinaison d'une normalisation des chiffres et une exclusion des mot-outils permet d'obtenir les meilleurs résultats. Elle donne une performance significativement meilleure que les autres méthodes de normalisation que nous avons utilisées. Nous avons obtenu par cette méthode, une MAP de $0,0027$ avec un k-means clustering de 1000 clusters.

En vue de constituer une ressource d'un corpus de spécialité, l'extraction des termes candidats nous permet de cibler les termes qui correspondent au mieux à notre objectif d'extraction d'information. Nous avons extrait une grande proportion de termes du domaine biomédical grâce à l'extracteur de terme `YaTeA`. L'analyse syntaxique proposée par cet outil permet de définir les mots cibles et les contextes.

L'analyse distributionnelle des mots est basée sur les contextes partagés par les mots cibles de telle sorte que le traitement de contextes est important. La normalisation des contextes dans une certaine mesure peut améliorer la performance. Pourtant exclure trop de stop-words risque de faire perdre des informations liées aux contextes et diminuer la performance. Ainsi dans nos expériences, la méthode d'exclusion d'une liste de mots-outils présente de meilleurs résultats que celle d'exclure une grande liste de stop-words.

L'analyse distributionnelle fournit un moyen efficace de regrouper des mots sémantiquement proches. Mais quant à la classification de mots, nous avons rencontré certaines restrictions avec le regroupement des termes candidats similaires et la classification. La méthode du `word2vec` ne permet pas de traiter la polysémie et l'hyponymie, les mots qui partagent les mêmes contextes ne sont pas forcément d'une même classe. La réalisation de la classification des termes est en effet difficile.

6.2 Perspectives

Pour régler la difficulté et les points faibles que nous avons mentionnés, nous prévoyons en perspective, pour notre méthode, quelques travaux d'amélioration.

Normalisation des variations de termes Nous constatons dans notre corpus qu'il existe des termes qui présentent différentes formes. Ces termes qui ont plusieurs variations sont traités comme des différents mots dans la représentation des mots par `word2vec`. Il nous faudra faire une normalisation à l'aide une ressource terminologique comme par exemple, UMLS ou une méthode automatique [Perinet, 2015].

Algorithme de catégorisation L'algorithme de k-means clustering que nous avons utilisé pour la classification des termes candidats oblige à choisir le paramètre k. Le choix de k est assez libre et difficile à fixer. Il nous faudra donc explorer d'autres méthodes de classification, comme le regroupement hiérarchique utilisé dans le travail de [Alfalahi et al., 2015].

BIBLIOGRAPHIE

[Alfalahi et al., 2015] Alfalahi, A., Ahlblom, R., Skeppstedt, M., Baskalayci, R., Henriksson, A., Asker, L., Paradis, C., and Kerren, A. (2015). Expanding a dictionary of marker words for uncertainty and negation using distributional semantics. pages 90–96. – Cité page 44.

[Aubin and Hamon, 2006] Aubin, S. and Hamon, T. (2006). Improving term extraction with terminological resources. In Salakoski, T., Ginter, F., Pyysalo, S., and Pahikkala, T., editors, *Advances in Natural Language Processing (5th International Conference on NLP, FinTAL 2006)*, number 4139 in LNAI, pages 380–387. Springer. – Cité page 18.

[Bengio et al., 2006] Bengio, Y., Schwenk, H., Senécal, J.-S., Morin, F., and Gauvain, J.-L. (2006). Neural probabilistic language models. In *Innovations in Machine Learning*, pages 137–186. Springer. – Cité page 26.

[Bodenreider et al., 2002] Bodenreider, O., Rindflesch, T. C., and Burgun, A. (2002). Unsupervised, corpus-based method for extending a biomedical terminology. In *Proceedings of the ACL-02 workshop on Natural language processing in the biomedical domain-Volume 3*, pages 53–60. Association for Computational Linguistics. – Cité pages 9 et 12.

[Cabré Castellví et al., 2001] Cabré Castellví, M. T., Estopà Bagot, R., and Vivaldi Palatresi, J. (2001). Automatic Term Detection: a review of current systems. *Recent Advances in Computational Terminology*, (2001):53–88. – Cité page 12.

[Cohen and Hersh, 2005] Cohen, A. M. and Hersh, W. R. (2005). A survey of current work in biomedical text mining. *Briefings in bioinformatics*, 6(1):57–71. – Cité page 11.

[Cohen and Demner-Fushman, 2014] Cohen, K. B. and Demner-Fushman, D. (2014). *Biomedical natural language processing*, volume 11. John Benjamins Publishing Company. – Cité page 9.

[Cohen and Hunter, 2008] Cohen, K. B. and Hunter, L. (2008). Getting started in text mining. *PLoS Comput Biol*, 4(1):e20. – Cité page 9.

[Collobert et al., 2011] Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537. – Cité page 26.

[Cook, 1989] Cook, V. (1989). *The relevance of grammar in the applied linguistics of language teaching*. Trinity College, Centre for Language and Communication Studies. – Cité pages 29 et 37.

[Curran, 2004] Curran, J. R. (2004). From distributional to semantic similarity. – Cité page 10.

[Grabar and Zweigenbaum, 2004] Grabar, N. and Zweigenbaum, P. (2004). Lexically-based terminology structuring. *Terminology*, 10(1):23–53. – Cité pages 26 et 27.

[Grishman and He, 2014] Grishman, R. and He, Y. (2014). An information extraction customizer. In *Text, Speech and Dialogue*, pages 3–10. Springer. – Cité pages 10 et 12.

[Hamon et al., 2007] Hamon, T., Nazarenko, A., Poibeau, T., Aubin, S., and Derivière, J. (2007). A robust linguistic platform for efficient and domain specific web content analysis. In *Proceedings of RIAO 2007*, Pittsburgh, USA. 15 pages. – Cité page 18.

[Harris, 1954] Harris, Z. S. (1954). Distributional structure. *Word*. – Cité page 25.

[Henriksson et al., 2011] Henriksson, A., Hassel, M., and Kvist, M. (2011). Diagnosis code assignment support using random indexing of patient records–a qualitative feasibility study. In *Artificial Intelligence in Medicine*, pages 348–352. Springer. – Cité page 12.

[Hersh et al., 1996] Hersh, W. R., Campbell, E. H., Evans, D. A., and Brownlow, N. D. (1996). Empirical, automated vocabulary discovery using large text corpora and advanced natural language processing tools. In *Proceedings of the AMIA Annual Fall Symposium*, page 159. American Medical Informatics Association. – Cité page 12.

[Kim et al., 2003] Kim, J.-D., Ohta, T., Tateisi, Y., and Tsujii, J. (2003). Genia corpus—a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(suppl 1):i180–i182. – Cité pages 8, 15, 17 et 18.

[Kim et al., 2004] Kim, J.-D., Ohta, T., Tsuruoka, Y., Tateisi, Y., and Collier, N. (2004). Introduction to the bio-entity recognition task at jnlpba. In *Proceedings of the international joint workshop on natural language processing in biomedicine and its applications*, pages 70–75. Association for Computational Linguistics. – Cité page 16.

[Mandel, 2006] Mandel, M. A. (2006). Integrated annotation of biomedical text: creating the pennbioie corpus. *Text Mining Ontologies and Natural Language Processing in Biomedicine, Manchester, UK*. – Cité page 18.

[McCray et al., 2002] McCray, A. T., Browne, A. C., and Bodenreider, O. (2002). The lexical properties of the gene ontology. In *Proceedings of the AMIA Symposium*, page 504. American Medical Informatics Association. – Cité pages 9 et 11.

[Meystre et al., 2008] Meystre, S. M., Savova, G. K., Kipper-Schuler, K. C., Hurdle, J. F., et al. (2008). Extracting information from textual documents in the electronic health record: a review of recent research. *Yearb Med Inform*, 35:128–44. – Cité pages 9 et 11.

[Mikolov et al., 2013a] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*. – Cité pages 12, 26 et 30.

[Mikolov et al., 2013b] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119. – Cité page 26.

[Mikolov et al., 2013c] Mikolov, T., Yih, W.-t., and Zweig, G. (2013c). Linguistic regularities in continuous space word representations. In *HLT-NAACL*, pages 746–751. – Cité page 12.

[Miñarro-Giménez et al., 2015] Miñarro-Giménez, J. A., Marín-Alonso, O., and Samwald, M. (2015). Applying deep learning techniques on medical corpora from the world wide web: a prototypical system and evaluation. *arXiv preprint arXiv:1502.03682*. – Cité page 12.

[Muneeb et al., 2015] Muneeb, T., Sahu, S. K., and Anand, A. (2015). Evaluating distributed word representations for capturing semantics of biomedical concepts. *ACL-IJCNLP 2015*, page 158. – Cité pages 12 et 29.

[Pazienza et al., 2005] Pazienza, M. T., Pennacchiotti, M., and Zanzotto, F. M. (2005). Terminology extraction: an analysis of linguistic and statistical approaches. In *Knowledge Mining*, pages 255–279. Springer. – Cité pages 10 et 12.

[Pennington et al., 2014] Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. *Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014)*, 12:1532–1543. – Cité page 12.

[Perinet, 2015] Perinet, A. (2015). Analyse distributionnelle appliquée aux textes de spécialité: réduction de la dispersion des données par abstraction des contextes. – Cité pages 26, 30 et 44.

[Řehůřek and Sojka, 2010] Řehůřek, R. and Sojka, P. (2010). Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA. http://is.muni.cz/publication/884893/en. – Cité page 26.

[Sahlgren, 2005] Sahlgren, M. (2005). An introduction to random indexing. In *Methods and applications of semantic indexing workshop at the 7th international conference on terminology and knowledge engineering, TKE*, volume 5. – Cité pages 12 et 26.

[Swanson, 1988] Swanson, D. R. (1988). Migraine and magnesium: eleven neglected connections. *Perspectives in biology and medicine*, 31(4):526–557. – Cité page 11.

[Tsuruoka et al., 2005] Tsuruoka, Y., Tateishi, Y., Kim, J.-D., Ohta, T., McNaught, J., Ananiadou, S., and Tsujii, J. (2005). Developing a robust part-of-speech tagger for biomedical text. In *Proceedings of Advances in Informatics - 10th Panhellenic Conference on Informatics*, LNCS 3746, pages 382–392. – Cité page 18.

[Zweigenbaum et al., 2007] Zweigenbaum, P., Demner-Fushman, D., Yu, H., and Cohen, K. B. (2007). Frontiers of biomedical text mining: current progress. *Briefings in bioinformatics*, 8(5):358–375. – Cité page 11.