Adaptation et combinaison d'approches bio-inspirées et de fouille de textes pour la sélection de descripteurs textuels

Nourelhouda Yahi

République Algérienne Démocratique et Populaire
Ministère de l'enseignement supérieur et de la recherche scientifique
Université Abdelhamid Mehri Constantine 2

Faculté des Nouvelles Technologies de l'Information et de la Communication-NTIC
Département d'informatique fondamentale et ses applications-IFA

Année: .....
N° d'ordre: .....
N° de série: .....

THÈSE

Pour l'obtention du diplôme de Doctorat 3ème cycle LMD
Spécialité : Informatique

Adaptation et combinaison d'approches bio-inspirées et de fouille de textes pour la sélection de descripteurs textuels

---

Présentée par

Mme YAHI Nourelhouda

Soutenue le .25 / 02 / 2021. devant le jury:

Pr. ZAROUR NACEREDDINE	Université Abdelhamid MEHRI - Constantine 2	Président
Pr. BELHADEF HACENE	Université Abdelhamid MEHRI - Constantine 2	Encadreur
Dr. MATHIEU ROCHE	UMR TETIS, CIRAD Montpellier, France	Co-encadreur
Dr. MOSTEFAI SIHAM	Université Abdelhamid MEHRI - Constantine 2	Examinatrice
Dr. BOUAKKAZ MUSTAPHA	Université Amar TELIDJI de Laghouat	Examinateur
Dr. BOUKRAA DOULKIFLI	Université Mohamed Seddik BENYAHIA de Jijel	Examinateur

Je dédie cette thèse
A mes chères parents

Nour...

Remerciement

J'adresse toute ma reconnaissance à mon directeur de thèse, Pr. Hacene Belhadef qui a dirigé ce travail de recherche. Je le remercie pour la confiance qu'il m'a accordée, sa gentillesse, sa disponibilité et son soutien durant toutes ces années de thèse.

Je remercie vivement Pr. Mathieu Roche, co-directeur de thèse, pour son aide, ses conseils et ses remarques pertinentes. Je tiens ici à lui exprimer ma gratitude et mon profond respect.

Je remercie également tous les Professeurs membres du jury. Qu'ils veuillent trouver ici toute ma reconnaissance pour avoir accepté l'examination et l'évaluation de ce travail.

Mention spéciale à mon époux, mes enfants et tous les membres de ma famille qui m'ont soutenu et accompagné durant mes années d'étude, ainsi que mes amies, mes chers et mes collègues qui, par leur support et encouragement, m'ont permis de m'investir entièrement dans mes études.

Je remercie tout le personnel du laboratoire MISC sans oublier tous le personnel du département d'informatique de l'université Abdelhamid Mehri Constantine 2.

Je tiens à remercier toutes les personnes qui, directement ou indirectement ont contribué à la réalisation de ce travail.

Nourlhouda Yahi

ملخص

يعتبر تشابه النص الدلالي حجر الزاوية في فهم النص وعنصراً مهماًا للعديد من مهام معالجة اللغة الطبيعية. مبدأها على مجموعة من البيانات النصية هو تحديد وجود تشابه دلالي أو لتقييم درجة التشابه بينهما ، هذه البيانات النصية التي تم إنشاؤها من مصادر مختلفة هي أمثلة على البيانات غير المهيكلة. لا تتناسب البيانات غير المهيكلة تماماًا مع الهيكل التقليدي لقواعد البيانات العلائقية ، فهي معقدة ويصعب التعامل معها وتتطلب خطوة إعداد ، ويسمح لنا هذا الإعداد بتوليد أوصاف نصية أفضل ، والتي تقودنا إلى معالجة فعالة.

الهدف من هذه الأطروحة هو تقديم حالة من الفن حول طرق المراحل الثلاث لإعداد البيانات: المعالجة المسبقة ، وتمثيل المتجه واختيار الخصائص ، من خلال تحليل تأثيرها على مهمة تقييم التشابه الدلالي بين الكيانات النصية ، لنفس الغرض ، تم اقتراح مناهج مستوحاة من الأحياء فعالة وبسيطة وقوية. في إطار هذه الأطروحة ، نقترح مساهمات مختلفة. المساهمة الأولى هي نهج قائم على استخلاص الواصفات اللغوية من نص ومصطلحات خاصة بقاموس المرادفات من خلال تطبيق ترجيح دلالي محدد. المساهمة الثانية هي نهج غير خاضع للإشراف يعتمد على مزيج من الأساليب المستوحاة من الحيوية والتنقيب عن النص من أجل بحث فعال عن المجموعات الفرعية المثلى لخصائص المستندات النصية ، وتكمن مساهمة هذا النهج في تكييف خوارزمية جينية مستوحاة من الكم. . تمثل المساهمة الثالثة نسخة خاضعة للإشراف من النهج المستوحى من الحيوية المقترحة بالفعل ، أثناء دراسة تأثير تقنيات المعالجة المسبقة المستخدمة على نطاق واسع على مهمة التشابه الدلالي. ننتهي من خلال دمج تقنيات تضمين المستندات كطرق تمثيل البيانات ، مع تقييم تأثير المعالجة المسبقة على هذه الأساليب. يتم إجراء مقارنة تجريبية ، مع الأخذ في الاعتبار التشابه الدلالي كدراسة حالة.

تم اختبار هذه المقترحات على بيانات مكونة من مصادر مختلفة ومجموعات بيانات قياسية. أثبتت النتائج التي تم الحصول عليها فعالية الطرق المقترحة.

الكلمات المفتاحية معالجة اللغة االطبيعية ، التشابه الدلالي ، المعالجة المسبقة ، تمثيل المتجهات ، اختيار الميزات ، تضمين المستندات ، الخوارزمية الجينية المستوحاة من الكم.

Abstract

Semantic textual similarity is considered to be the backbone of text understanding and an important element for many natural language processing tasks. Its principle on a set of textual data is to identify the existence of a semantic similarity or to assess the degree of similarity between them, these textual data generated from different sources are examples of unstructured data. Unstructured data does not fit perfectly into the traditional structure of relational databases, they are complicated and difficult to handle and require a preparation step, this preparation allows us to generate better textual descriptors, which they lead us to effective treatment.

The objective of this thesis is to present a state of the art on the methods of the three phases of data preparation : The pre-processing, the vector representation and the features selection, by analyzing their impact on the task of semantic similarity evaluation between textual entities, for the same purpose, efficient, simple and robust bio-inspired approaches have been proposed. Within the framework of this thesis, we propose different contributions. The first contribution is an approach based on the extraction of linguistic descriptors from a text and terms specific to a thesaurus by applying a specific semantic weighting. The second contribution is an unsupervised approach based on a combination of bio-inspired approaches and text mining for an efficient search of the optimal subsets of the characteristics of text documents, the contribution of this approach lies in the adaptation of a quantum inspired genetic algorithm. The third contribution represents a supervised version of the already proposed bio-inspired approach, while examining the impact of widely used preprocessing techniques on the semantic similarity task. We finish by integrating "Document Embedding" techniques as data representation methods, while evaluating the impact of preprocessing on these methods. An empirical comparison is made, taking semantic similarity as a case study.

These proposals were tested on data made up of different sources or standard datasets. The results obtained proved the effectiveness of the proposed methods.

Keywords Natural language processing, semantic similarity, preprocessing, Vector representation, Feature selection, Word embedding, Quantum inspired genetic algorithm.

Résumé

La similarité textuelle sémantique est considérée comme la pierre angulaire de la compréhension des textes et un élément important pour de nombreuses tâches de traitement du langage naturel. Son principe sur un ensemble de données textuelles est d'identifier l'existence d'une similarité sémantique ou d'évaluer le degré de similarité entre eux, ces données textuelles générées à partir de différentes sources sont des exemples de données non structurées. Les données non structurées ne s'intègrent pas parfaitement dans la structure traditionnelle des bases de données relationnelles, elles sont compliquées et difficiles à manipuler et nécessitent une étape de préparation, cette préparation permet de générer des meilleurs descripteurs textuels, qu'ils nous conduisent à un traitement efficace.

L'objectif de cette thèse est de présenter un état de l'art sur les méthodes des trois phases de préparation des données : Le pré-traitement, la représentation vectorielle et la sélection des caractéristiques, en analysant leur impact sur la tâche de l'évaluation de similarité sémantiques entre entités textuelles, dans le même but, des approches bio-inspirées efficaces, simples et robustes ont été proposé. Dans le cadre de cette thèse, nous proposons différentes contributions. La première contribution est une approche fondée sur l'extraction de descripteurs linguistiques issus d'un texte et des termes propres à un thésaurus en appliquant une pondération sémantique spécifique. La deuxième contribution est une approche non-supervisée basée sur une combinaison d'approches bio-inspirées et de fouille de textes pour une recherche efficace des sous-ensembles optimales des caractéristiques des documents texte, la contribution de cette approche se réside dans l'adaptation d'un algorithme génétique inspiré du quantique. La troisième contribution représente une version supervisée de l'approche bio-inspirée déjà proposée, tout en examinant l'impact des techniques de pré-traitement largement utilisées sur la tâche de similarité sémantique. On termine par l'intégration des techniques de plongement de documents comme méthodes de représentation des données, tout en évaluant l'impact de pré-traitement sur ces méthodes. Une comparaison empirique est réalisée, en prenant la similarité sémantique comme étude de cas.

Ces propositions ont été expérimentées sur des données constituées des sources différentes et des datasets standards. Les résultats obtenus ont prouvé l'efficacité des méthodes proposées.

Mots-clés Traitement de langage naturel, Similarité sémantique, pré-traitement, Représentation vectorielle, Sélection de caractéristique, Plongement de mots, Algorithme génétique inspiré du quantique.

Table des matières

Table des figures

Liste des tableaux

Introduction Générale

Introduction

Le traitement automatique du langage naturel (TALN) ou ce que l'on appelle en anglais Natural Language Processing (NLP) est un domaine intégral de l'informatique dans lequel l'apprentissage automatique et la linguistique informatique sont largement utilisés. Ce domaine vise principalement à rendre l'interaction humaine et informatique facile mais efficace. La machine apprend la syntaxe et la signification du langage humain, le traite et donne la sortie à l'utilisateur. Le domaine du NLP consiste à créer des systèmes informatiques pour des tâches significatives avec un langage compréhensible naturel et humain. Parmi les différentes tâches du NLP, on pourrait citer par exemple la similarité sémantique qui est considérée comme la colonne vertébrale de la compréhension des textes et aussi un élément important pour de nombreuses tâches telles que la synthèse de documents, la clarification du sens des mots, la notation des réponses courtes, la recherche et l'extraction d'informations.

La similarité sémantique peut être définie par une métrique sur un ensemble de données textuelles avec l'idée est de trouver la similarité sémantique entre eux, ces données textuelles générées à partir de conversations, de déclarations ou même des réseaux sociaux sont des exemples de données non structurées. Les données non structurées ne s'intègrent pas parfaitement dans la structure traditionnelle de lignes et de colonnes des bases de données relationnelles et représentent la grande majorité des données disponibles dans le monde réel. C'est compliqué et difficile à manipuler ces données incomplètes, bruyantes et inconsistantes qui nécessitent une préparation pour qu'elles soient prêtes pour un traitement efficace.

Problématique et Motivation

Dans l'apprentissage automatique, la préparation des données est le processus d'initialisation des données pour l'apprentissage, les tests et la mise en œuvre d'un algorithme d'apprentissage. Il s'agit d'un processus en plusieurs étapes qui implique la collecte, le nettoyage, le pré-traitement des données et l'ingénierie des caractéristiques. Ces étapes jouent un rôle important dans la qualité globale de modèle d'apprentissage automatique, car elles s'appuient les unes sur les autres pour garantir que le modèle répond aux attentes.

La préparation des données peut être l'une des étapes les plus difficiles de tout projet d'apprentissage automatique. La raison est que chaque ensemble de données est différent et très spécifique au projet. Néanmoins, il existe suffisamment de points communs entre les projets de modélisation prédictive pour que nous puissions définir une séquence d'étapes génériques à mettre en oeuvre.

Ce processus souffre de certains problèmes qui peuvent dégrader les performances des systèmes, ces problèmes peuvent résider dans la phase de pré-traitement, de représentation vectorielle ou de sélection des caractéristiques. Dans le cadre du projet de cette thèse, nous nous intéressons à la résolution des différents problèmes rencontrés par la proposition de nouvelles approches d'identification des descripteurs afin d'améliorer la performance des systèmes d'évaluation de similarité sémantique.

— Représentation vectorielle : L'étape la plus fondamentale pour la majorité des tâches de traitement du langage naturel est de permettre aux ordinateurs de comprendre, d'analyser et d'extraire du sens du langage naturel humain, et cela de manière intelligente et utile. En ce qui concerne les données textuelles,

les ordinateurs sont plus aptes à gérer les valeurs numériques que du texte réel transmet sous forme de "mots". La tâche de transformation des mots en vecteurs de nombres qui représentent leur signification s'appelle la représentation vectorielles de texte. Le grand défi dans cette tâche est de transférer autant d'informations que possible sur les données textuelles tout en les convertissant en données numériques, telles que la cooccurrence entre les mots, l'ordre des mots dans le texte, la sémantique des entités textuelles, etc. Dans le cadre de cette thèse nous nous intéressons à la proposition des nouvelles méthodes de pondération, ainsi que l'intégration des méthodes de représentation des données qui ont prouvé leur impact positif sur un certain nombre d'applications, et cela afin d'améliorer l'exactitude d'évaluation de similarité sémantique des documents textes.

— Sélection des caractéristiques : C'est une étape permettant la recherche de l'ensemble minimum de caractéristiques idéalement nécessaires et suffisants pour décrire un ensemble de documents textuels. La qualité des systèmes de traitement du langage naturel dépend directement du bon choix du contenu des vecteurs de caractéristiques. Ces vecteurs peuvent contenir des caractéristiques redondantes et insignifiantes, ces dernières peuvent conduire à un problème de complexité spatio-temporelle et une diminution au niveau de performance de système. Dans le cadre de cette thèse, nous nous intéressons également à la résolution du problème de sélection et de réduction de caractéristiques en utilisant des approches bio-inspirées.

Contributions

Dans ce travail de thèse, nous avons proposé plusieurs contributions afin d'élaborer des méthodes permettant d'extraire des descripteurs textuels optimaux pouvant être utilisés dans les tâches de fouille de textes (Text-Mining). Notre objectif donc est d'élaborer des méthodes qui seraient à la fois simples et efficaces donnant des résultats de bonne qualité dans un temps raisonnable. Ainsi, quatre approches ont été proposées dans cette thèse.

— Une approche fondée sur l'extraction de descripteurs linguistiques issus d'un texte et des termes propres à un thésaurus en appliquant une pondération sémantique spécifique. Notre méthode a tendance à rapprocher des textes ayant des thématiques proches ce qui permet de mettre en relation des données de qualité différente.

— Une approche non-supervisée basée sur un algorithme bio-inspiré pour mettre en correspondance des documents textes hétérogènes. La contribution de cette approche se réside dans l'adaptation d'un algorithme bio-inspirée pour une recherche efficace des sous-ensembles optimaux des caractéristiques des documents texte.

— Une version supervisée de l'approche bio-inspirée déjà proposée, tout en examinant l'impact des techniques de pré-traitement largement utilisées dans la tâche de similarité sémantique.

— Utilisation des techniques de plongement de documents comme méthodes de représentation des données, tout en évaluant l'impact de pré-traitement sur

ces méthodes. Une comparaison empirique est réalisée, en prenant la similarité sémantique comme étude de cas.

Organisation de la Thèse

Le travail présenté dans cette thèse est organisé en huit chapitres regroupés en deux parties, respectivement : État de l'art et contributions.

La première partie est constituée de quatre chapitres, décrivant l'état de l'art destiné à l'évaluation de similarité sémantique, le pré-traitement des données, la représentation des données et la sélection des caractéristiques. Dans le chapitre 1, nous introduisons les concepts clés du traitement du langage naturel et de la similarité sémantique, en décrivant son processus général et ses domaines d'application. Dans le chapitre 2, nous passons en revue les notions fondamentales liées au pré-traitement des données textuelles, ensuite nous détaillons les différentes techniques connues dans la littérature, ensuite nous citons les techniques adoptées pour l'évaluation de la similarité sémantique entre documents textes. Dans le chapitre 3, nous abordons le principe de la représentation des données textuelles. Par la suite, nous définissons les différentes stratégies de représentation vectorielle discrète et distribuée. Le dernier chapitre de cette première partie présente la sélection de caractéristiques, ses propriétés et ses différentes stratégies et méthodes.

La deuxième partie est dédiée à la présentation de nos contributions et expérimentations réalisées, elle est organisée en quatre chapitres. Dans le chapitre 5, nous proposons notre approche de mise en correspondance de données textuelles hétérogènes à partir de Thésaurus. Le chapitre 6 décrit notre approche bio-inspirée non-supervisée qui sert à la sélection des caractéristiques pour faire correspondre des documents textes. Le chapitre 7 représente sa version supervisée, une étude expérimentale sur l'impact de pré-traitement sur cette approche est également présentée. A la fin, le chapitre 8 présente une étude comparative sur l'impact de pré-traitement de les représentation distribuées, dans un cadre de l'évaluation de similarité sémantique entre documents textuels.

Nous achevons ce manuscrit par une conclusion générale qui résume les travaux menés dans le cadre de cette thèse ainsi que des éventuels perspectives à réaliser dans des futurs travaux.

Première partie

État de l'art

Chapitre 1

Similarité Sémantique

Sommaire

1.1 Introduction

Depuis les débuts de l'informatique, l'homme cherche à communiquer avec les machines. Si les nombreux langages de programmation permettent une forme d'échange entre l'homme et la machine, on aimerait que cette communication se fasse de façon plus naturelle. Pour que cela soit possible, il faut d'abord que la machine "comprenne" ce que l'utilisateur lui dit pour être capable de répondre d'une manière compréhensible par l'homme.

Largement utilisée dans les organisations axées sur les connaissances, la fouille de texte consiste à examiner de grandes collections de documents textuels pour découvrir de nouvelles informations ou aider à répondre à des questions de recherche spécifiques [119]. L'exploration de texte identifie des faits, des relations et des affirmations qui, autrement, resteraient enfouis dans la masse des mégadonnées textuelles. Une fois extraites, ces informations sont converties en une forme structurée qui peut être analysée plus avant ou présentée directement à l'aide de tableaux groupés, de cartes mentales, de graphiques, etc. L'exploration de texte utilise une variété de méthodologies pour traiter le texte, l'une des plus importantes étant le traitement du langage naturel qui repose sur la combinaison de méthodes linguistiques et statistiques.

Des concepts de base et des définitions sous-jacents à notre travail de thèse sont présentés dans la section1.2. La section 1.3 donne une description des tâches de similarité sémantique. La section 1.4 présente quelque stratégies et méthodes pour l'évaluation de similarité sémantique. Enfin, nous fournissons une conclusion dans la section 1.5.

1.2 Concepts de base

Une partie des travaux en traitement du langage naturel reposent sur des méthodes de l'intelligence artificielle en particulier pour les tâches qui s'intéressent à la représentation des connaissances et au raisonnement.

Le traitement du langage naturel aide les ordinateurs à comprendre, interpréter et manipuler le langage humain, elle utilise une variété de méthodologies pour déchiffrer les ambiguïtés dans le langage humain [42]. Cette discipline, comme le montre la figure 1.1, s'inspire de nombreuses autres disciplines, notamment l'apprentissage automatique, l'apprentissage profond et les approches linguistiques, dans le but de combler le fossé entre la communication humaine et la compréhension informatique.

Dans cette section, nous présentons les concepts de base sous-jacents à notre travail de thèse.

1.2.1 Intelligence artificielle

L'intelligence artificielle ou ce que l'on appelle en anglais Artificial Intelligence (AI) fait référence à la simulation de l'intelligence humaine dans des machines programmées pour penser comme des humains et imiter leurs actions [94]. Ses points clés à retenir :

— Cette discipline fait référence à la simulation de l'intelligence humaine dans les machines.

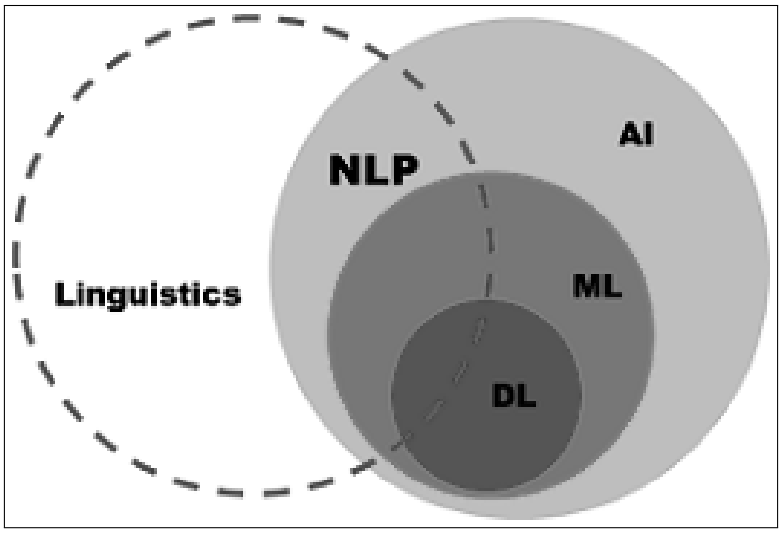

FIGURE 1.1: Relation entre NLP et les autres domaines

— Ses objectifs comprennent l'apprentissage, le raisonnement et la perception.
— Elle est utilisée dans différents secteurs, notamment la finance et la santé.
— Une AI faible a tendance à être simple et orientée vers une seule tâche, tandis qu'une AI forte exécute des tâches plus complexes et plus humaines.

1.2.2 Apprentissage automatique

L'apprentissage automatique ou ce que l'on appelle en anglais Machine Learning (ML) représente l'un des développements les plus prolifiques de l'intelligence artificielle moderne. Il fournit une nouvelle génération de techniques et d'outils de calcul qui soutiennent la compréhension et l'extraction de connaissances utiles à partir d'ensembles de données complexes [83]. Alors, qu'est-ce que l'apprentissage automatique ?

Simon [149] a défini l'apprentissage automatique comme : "Les changements dans le système qui sont adaptatifs en ce sens qu'ils permettent au système d'accomplir la même tâche ou des tâches tirées de la même population plus efficacement la prochaine fois." Par conséquent, fondamentalement, l'apprentissage automatique met l'accent sur la capacité du système à s'adapter ou à changer. En règle générale, c'est en réponse à une certaine forme d'expérience fournie au système. Après apprentissage ou adaptation, le système devrait avoir de meilleures performances futures sur la même tâche ou sur une tâche connexe [156].

Classiquement, il existe trois paradigmes d'apprentissage de base, à savoir l'apprentissage supervisé, l'apprentissage non supervisé et l'apprentissage par renforcement.

Dans l'apprentissage supervisé, l'apprenant reçoit un ensemble d'entrées avec les sorties souhaitées correspondantes. C'est similaire au processus d'apprentissage humain familier pour la reconnaissance de formes, dans lequel un enseignant fournit des exemples pour apprendre aux enfants à reconnaître différents objets (par exemple, des animaux). Une telle tâche de reconnaissance de formes est caractérisée par des données avec chaque échantillon d'entrée associé à une étiquette, à savoir des données étiquetées.

Contrairement à l'apprentissage supervisé, les tâches d'apprentissage non supervisé sont caractérisées par des données qui consistent uniquement en des entrées, à savoir que les données ne sont pas étiquetées et qu'il n'y a plus la présence d'un enseignant. L'apprentissage non supervisé vise à trouver certaines structures de dépendance sous-jacentes aux données via l'optimisation d'un principe d'apprentissage.

Le troisième paradigme est l'apprentissage par renforcement. En observant l'environnement actuel et en obtenant des commentaires (le cas échéant), l'apprenant effectue une action et change un nouvel environnement, recevant une valeur d'évaluation (récompense ou punition) sur l'action. Un processus d'apprentissage fait une série d'actions avec le prix total reçu maximisé. Contrairement à l'apprentissage non supervisé, l'apprenant est guidé par une évaluation externe. Contrairement à l'apprentissage supervisé dans lequel l'enseignant spécifie clairement le résultat qui correspond à une entrée, dans l'apprentissage par renforcement, l'apprenant ne reçoit qu'une valeur évaluative sur l'action réalisée [83].

1.2.3 Apprentissage profond

L'apprentissage profond ou ce que l'on appelle en anglais Deep Learning (DL) est un mot à la mode aujourd'hui. Cependant, il y a un manque de définition unifiée de l'apprentissage profond dans la littérature.

Selon Deng [44] "DL est une classe d'algorithmes de ML qui : (1) utilisent une cascade de plusieurs couches d'unités de traitement non linéaires pour l'extraction et la transformation de caractéristiques. Chaque couche successive utilise la sortie de la couche précédente comme entrée, (2) apprend plusieurs niveaux de représentations qui correspondent à différents niveaux d'abstraction ; les niveaux forment une hiérarchie de concepts."

Selon Schmidhuber [145], "DL (également appelé apprentissage structuré profond ou apprentissage hiérarchique) consiste à apprendre des représentations de données, par opposition à des algorithmes spécifiques à une tâche. L'apprentissage peut être supervisé, semi-supervisé ou non supervisé."

Selon Marshall [67] "DL est un sous-ensemble de ML qui dispose d'un niveau hiérarchique de réseaux de neurones artificiels capables d'apprendre sans surveillance à partir de données non structurées ou non étiquetées pour mener à bien le processus de ML. Également appelé apprentissage neuronal profond ou réseau neuronal profond, l'adjectif «profond» provient de l'utilisation de plusieurs couches dans le réseau."

1.2.4 Linguistiques

Une discipline qui sert à analyser, synthètiser et comprendre la langue écrite et parlée. Une compréhension informatique du langage fournit aux êtres humains un aperçu de la pensée et de l'intelligence. Les ordinateurs qui sont linguistiquement compétents aident non seulement à faciliter l'interaction humaine avec les machines et les logiciels, mais rendent également les ressources textuelles et autres de l'internet facilement disponibles dans plusieurs langues [141].

1.2.5 Traitement du Langage Naturel

Le Traitement du Langage Naturel ou ce que l'on appelle en anglais Natural Language Processing NLP est une gamme de techniques informatiques motivées par la théorie pour analyser et représenter des textes naturels à un ou plusieurs niveaux d'analyse linguistique dans le but de réaliser un traitement du langage de type humain pour une gamme de tâches ou d'applications [106]. Plusieurs éléments de cette définition peuvent être détaillés.

— Premièrement, la notion imprécise de «gamme de techniques de calcul» est nécessaire car il existe plusieurs méthodes ou techniques parmi lesquelles nous choisissons, pour réaliser un type particulier d'analyse linguistique.

— Les «textes naturels» peuvent être de n'importe quelle langue, mode, genre, etc. Les textes peuvent être oraux ou écrits. La seule exigence est qu'ils soient dans une langue utilisée par les humains pour communiquer entre eux. En outre, le texte analysé ne doit pas être spécifiquement construit aux fins de l'analyse, mais plutôt que le texte soit recueilli à partir de l'usage réel.

— La notion de «niveaux d'analyse linguistique» fait référence au fait qu'il existe plusieurs types de traitement du langage connus pour fonctionner lorsque les humains produisent ou comprennent le langage, par exemple : lexical, syntaxique, sémantique, etc. On pense que les humains utilisent normalement tous ces niveaux puisque chaque niveau véhicule différents types de sens. Mais divers systèmes NLP utilisent différents niveaux, ou combinaisons de niveaux d'analyse linguistique, et cela se voit dans les différences entre les diverses applications NLP.

— Le «traitement du langage de type humain» révèle que le NLP est considérée comme une discipline au sein de l'AI. Et bien que la lignée complète du NLP dépend d'un certain nombre d'autres disciplines, puisque elle vise des performances de type humain, il convient de la considérer comme une discipline de l'AI.

— «Pour une gamme de tâches ou d'applications» souligne que le NLP n'est généralement pas considéré comme un objectif en soi, sauf peut-être pour les chercheurs en AI. Pour d'autres, le NLP est le moyen d'accomplir une tâche particulière. Par conséquent, on dispose de systèmes de recherche d'informations qui utilisent le NLP, ainsi que la traduction automatique, la réponse aux questions, etc.

Aujourd'hui, le NLP est en plein essor grâce aux énormes améliorations de l'accès aux données et à l'augmentation de la puissance de calcul, qui permettent aux praticiens d'obtenir des résultats significatifs dans des domaines tels que la santé [159], les médias [123] et la finance [88] entre autres. Il existe différentes tâches dans le NLP comme l'analyse des sentiments, la traduction automatique, la réponse aux questions, le résumé automatique et la similarité sémantique [98].

1.3 Similarité sémantique

En général, la similarité sémantique vise à évaluer un score qui représente la ressemblance entre les significations des entités comparées. L'évaluation de la similarité sémantique entre les entités textuelles est considérée comme la colonne vertébrale de

la compréhension des textes, elle peut être définie par une métrique sur un ensemble de documents textuels avec l'idée de trouver la similarité sémantique entre eux.

Les deux tâches les plus importantes de la similarité sémantique sont l'identification de paraphrases et la similarité sémantique textuelle.

1.3.1 Identification de paraphrases

L'identification de paraphrases ou ce que l'on appelle en anglais Paraphrase Identification (PI) concerne la capacité d'identifier des expressions linguistiques alternatives de même sens à différents niveaux textuels (niveau document, niveau paragraphe, niveau phrase, niveau mot ou combinaison entre eux) [24]. En résumé, son but est de déterminer si une paire de phrases a la même signification. Bien que PI soit un domaine de recherche actif et qu'il ait des applications possibles dans l'extraction d'informations, la traduction automatique, la recherche d'informations, l'identification automatique de la violation du droit d'auteur et la réponse aux questions [126]. Le tableau 1.1 montre un exemple des paires de textes avec leur identification '1' pour paraphrase et '0' pour non-paraphrase.

Paires de textes	**Équivalence sémantique**
PCCW's chief operating officer, Mike Butcher, and Alex Arena, the chief financial officer, will report directly to Mr So. Current Chief Operating Officer Mike Butcher and Group Chief Financial Officer Alex Arena will report to So.	1
A tropical storm rapidly developed in the Gulf of Mexico Sunday and was expected to hit somewhere along the Texas or Louisiana coasts by Monday night. A tropical storm rapidly developed in the Gulf of Mexico on Sunday and could have hurricane-force winds when it hits land somewhere along the Louisiana coast Monday night.	0

TABLE 1.1: Exemple de paires de textes avec leurs équivalences sémantiques, selon Microsoft Research Paraphrase Corpus [46]

1.3.2 Similarité sémantique textuelle

La similarité sémantique du texte ou ce que l'on appelle en anglais Semantic Textual Similarity (STS) consiste à mesurer le degré de similarité sémantique entre deux textes (documents, paragraphes, phrases ou combinaison de ceux-ci) [24, 4]. Pour ne pas confondre, PI donne une décision oui / non et STS identifie le degré d'équivalence des paires de textes et les note sur la base de leurs relations sémantiques. Le tableau 1.2 montre un exemple des paires de textes avec leur degré de similarité, allant de '0' pour les paires différents à '5' pour les paires similaires.

Paires de textes	**Remarque**	**Degré d'équivalence sémantique**
* A man is playing a musical keyboard. * A man is playing a keyboard piano.	Les deux phrases sont complètement équivalentes, car elles signifient la même chose.	5
* A woman is slicing some tofu. * A woman is cutting a block of tofu into small cubes.	Les deux phrases sont généralement équivalentes, mais diffèrent dans certains détails insignifiants.	4
* A group of people are dancing. * Women are dancing outside.	Les deux phrases sont à peu près équivalentes, mais certaines informations importantes sont différentes / manquantes.	3
* A man is playing a football. * A man is maneuvering a soccer ball with his feet.	Les deux phrases ne sont pas équivalentes, mais partagent certains détails.	2
* A person is slicing some onions. * A woman is chopping herbs.	Les deux phrases ne sont pas équivalentes, mais portent sur le même sujet.	1
* A train is moving. * A man is doing yoga.	Les deux phrases sont complètement différentes.	0

TABLE 1.2: Exemple de paires de textes avec leurs dégrées de similarité, selon Semantic Textual Similarity Benchmark [33]

1.4 Stratégies d'évaluation de similarité sémantique

Les mesures de similarité sémantique entre les textes peuvent être classées de la manière suivante :

1.4.1 Similarité topologique / basée sur la connaissance

Dans le domaine de la recherche d'information ou ce que l'on appelle en anglais Information Retrieval (IR), la recherche documentaire basée sur la similarité sémantique des mots a été largement étudiée et toutes ces méthodes prennent en compte les relations sémantiques et ontologiques qui existent entre les mots (ex : polysémie, synonyme etc.) [113]. Donc, sur la base de cette connaissance, la similarité sémantique entre les objets de l'ontologie peut être classée en trois groupes : basé sur les nœuds ; basé sur les bords ; et hybride où il combine les nœuds et les bords.

— Approche basée sur les nœuds / contenu d'information : les approches basées sur les nœuds ou le contenu d'information [139, 138], sont utilisées pour déterminer la similarité sémantique entre les concepts. Dans cette méthode, chacun des concepts ou noeuds pose la taxonomie «est-un» est conservé dans un ensemble appelé C et tous ces nœuds portent des concepts uniques. Intuitivement, l'une des clés de la similarité de deux concepts est celle avec laquelle ils partagent des informations en commun. Intuitivement, l'une des clés de la similarité de deux concepts est celle avec laquelle ils partagent des informations en commun [113]. En taxonomie, une relation directe entre deux concepts peut être trouvée par une méthode de comptage d'arêtes. Dans cette méthode, si

le chemin minimal entre deux nœuds est long, cela signifie qu'il est nécessaire d'aller plus haut dans la hiérarchie pour trouver une borne inférieure.

— Approche basée sur les bords / distance : L'approche basée sur les bords est le moyen direct de calculer la similarité sémantique dans la taxonomie. Il compte le nombre d'arêtes entre deux nœuds qui correspond aux concepts comparés [78]. Minimum le chemin entre deux nœuds, ils sont plus similaires.

— Approche hybride : Les méthodes basées sur les nœuds et les arêtes décrites dans les sections précédentes présentent de nombreuses différences entre elles. Les méthodes basées sur les bords semblent vraies sans raisonnement concis et, d'autre part, l'approche basée sur les nœuds semble plus précise que celle basée sur la distance. La mesure de distance a été relayée sur la connaissance subjective du réseau tandis que le WordNet [1] [118] a été utilisé non pas pour mesurer la similarité, mais pour la construction des couches du réseau [113].

Dans cette catégorie, on reconnaît les travaux de Fernando et Stevenson [56], qui ont utilisé des similarités au niveau des mots dérivées de la taxonomie WordNet. De même, Das et Smith [41] ont utilisé des grammaires de dépendances quasi-synchrones dans un modèle probabiliste incorporant WordNet. Contrairement à d'autres méthodes basées sur WordNet, Hassan [68] a suggéré une nouvelle approche appelée Analyse sémantique saillante qui utilisait la signification du contexte selon les liens de Wikipédia. Cette classe d'approches a obtenu une performance relativement subordonnée sur le dataset MicroSoft Research Paraphrase Corpus (MSRPC) [121].

1.4.2 Similarité statistique / basée sur le corpus

C'est un similarité statistique appris des données (c-à-d. Corpus), qui est une collection de texte écrit ou parlé. Dans cette méthode, un modèle statistique a d'abord été construit, puis la similarité est estimée [113]. Plusieurs modèles ont été proposés au cours des dernières années, parmi eux on cite les catégories suivantes :

— Analyse sémantique latente : Ce que l'on appelle en anglais Latent Semantic Analysis (LSA), dans cette méthode, les informations contextuelles des mots ont été extraites et représentées à partir d'un grand corpus de [101]. Dans la première étape, le texte est représenté comme une matrice dans laquelle les lignes et les colonnes représentent l'unique mots et segments de texte. Chaque entrée représente le nombre de fréquences du mot, qui apparaît dans le texte [102]. Les fréquences des cellules sont pondérées par une fonction, qui exprime l'importance d'un mot dans un texte et le degré de partage d'informations de type mot dans le domaine du discours.

— Analyse sémantique latente généralisée : Ce que l'on appelle en anglais Generalized Latent Semantic Analysis (GLSA), les performances de LSA se dégradent lorsque les vecteurs de mots sont générés à partir d'un corpus de texte, qui est de nature hétérogène [13]. Le cadre GLSA est utilisé pour trouver les termes et les vecteurs de document, sur la base de similarités de termes par paires motivés sémantiquement , au lieu des vecteurs de document, un sac de mots, qui est utilisé dans LSA. C'est un cadre dans lequel différentes mesures d'associations sémantiques de termes sont combinées avec différentes méthodes de réduction de dimensionnalité [142].

1. https ://wordnet.princeton.edu/

— Analyse sémantique explicite : Ce que l'on appelle en anglais Explicit Semantic Analysis (ESA), dans cette méthode, la signification de tout texte est représentée comme un vecteur pondéré de concepts basés sur Wikipedia [2]. À l'aide de techniques d'apprentissage automatique, des représentations de vecteurs ont été effectuées sur un espace de grande dimension. Cette méthode est utile pour les représentations sémantiques à grain fin de texte en langage naturel sans restriction [57]. Pour représenter le texte comme un mélange pondéré de concepts naturels, les articles de Wikipédia sont utilisés, car il s'agit d'une collection de la plus grande encyclopédie, qui est définie par les humains et peut être facilement expliqué. Des vecteurs d'interprétation sémantique ont été construits pour mapper les fragments de langage naturel à une séquence pondérée de concepts Wikipédia en fonction de leur ordre d'entrée. La similarité sémantique est calculée en comparant les vecteurs, en utilisant la métrique cosinus [171]. Cette analyse sémantique est de nature explicite, car la signification des concepts se fait sur la cognition humaine, plutôt des concepts latents utilisés en LSA.

— Information mutuelle ponctuelle - Recherche d'informations : Ce que l'on appelle en anglais Pointwise Mutual Information - Information Retrieval (PMI-IR) Il s'agit d'un algorithme d'apprentissage non supervisé, permettant de reconnaître le synonyme d'un mot problématique à partir d'un ensemble de mots alternatifs. Cet algorithme utilise n'importe quel moteur de recherche pour émettre une requête de recherche et analyser le résultat de la requête pour trouver le mot synonyme. L'algorithme d'apprentissage non supervisé utilise les informations mutuelles ponctuelles, pour analyser les statistiques des données, qui sont collectées par le moteur de recherche, c'est-à-dire la récupération d'informations [113]. Les performances de la méthode dépendent de deux choses : la puissance du langage de requête du moteur de recherche et l'indexation du moteur de recherche (c'est-à-dire la collection de documents).

  Les avantages d'une approche ML consistent en la capacité de rendre compte d'une grande masse d'informations et la possibilité d'incorporer différentes sources d'informations telles que morphologiques, syntaxiques, sémantiques entre autres en une seule exécution. Le principal obstacle à l'utilisation des techniques d'apprentissage automatique concerne la disponibilité des données d'entraînement [96].

L'application de stratégies entièrement ou substantiellement fondées sur des statistiques de corpus a permis de résoudre le problème d'évaluation de similarité sémantique [25, 77, 75]. Ji et Eisenstein [77] ont utilisé un modèle de similarité distributionnelle simple en concevant une métrique de pondération des termes discriminante appelée Term Frequency - Kullback Leibler Divergence (TF-KLD). Les auteurs ont affirmé que leur mesure nouvellement introduite surpasse le schéma de pondération TF-IDF largement utilisé (voir la section 3.2.3). Dans le même esprit, Blacoe et Lapata [25] ont utilisé trois représentations distributionnelles du texte : un espace sémantique simple, un espace sensible à la syntaxe et des embeddings de mots. Avec des niveaux de performance variables sur l'ensemble de données MSRPC, cette catégorie (basée sur Corpus) contient certaines des méthodes les plus performantes, notamment les travaux de Ji et Eisenstein [77] et Issa et al. [75].

2. http ://en.wikipedia.org

Méthode de similarité	**Approche**	**Avantages**	**Limites**
Méthodes basées sur le corpus	Utilisez un corpus pour obtenir la probabilité ou la fréquence d'un mot dans un corpus	Corpus prétraité pour réduire les calculs	1. Corpus dépend du domaine. 2. Certains mots peuvent avoir la même similarité. 3. Les vecteurs sémantiques sont rares.
Méthodes basées sur la connaissance	Utilisez des informations de dictionnaire telles que WordNet pour obtenir des similarités (par exemple, chemin et profondeur, relations de mots, etc.)	L'adoption d'une ontologie conçue par l'homme peut augmenter la précision	1. Mots limités. 2. Certains mots peuvent avoir la même similarité s'ils ont le même chemin et la même profondeur
Méthodes hybrides	Utilisez à la fois le corpus et les informations d'un dictionnaire.	Fonctionne généralement mieux	Calculs supplémentaires

TABLE 1.3: Comparaison des approches de similarité des mots et des phrases

1.5 Conclusion

Au cours de ce chapitre, nous avons présenté les concepts NLP ainsi que toutes les disciplines en relation, nous avons ensuite abordé la notion des similarité sémantique et les diverses sous-domaines associés et les différentes stratégies d'évaluation de similarité sémantique en citant leurs avantages et limites, ainsi quelques travaux de la littérature ont été entamé. Dans le chapitre suivant, nous nous intéressons aux outils de pré-traitement de données.

Chapitre 2

Pré-traitement des données textuelles

Sommaire

2.1 Introduction

Le pré-traitement du texte est l'étape qui précède la phase de la représentation vectorielles ; il permet de décomposer le texte en une forme prévisible et analysable pour les tâches de NLP [146, 99, 74].

Dans la pratique, on constate généralement que le pré-traitement des données représente la majorité des efforts d'ingénierie des données. L'importance et la popularité connues de cette tâche peuvent être soutenues par le fait que les données du monde réel peuvent être incomplètes, bruitées et incohérentes, ce qui peut impacter les modèles. Par ailleurs, la préparation des données peut générer un ensemble plus réduit, ce qui peut considérablement améliorer l'efficacité de l'exploration de données. Enfin, le pré-traitement des données génère des données de qualité, qui conduisent à des modèles plus adaptés [170].

La section 2.2 représente quelques techniques de pré-traitements des données textuelles. La section 2.3 représente un état de l'art sur les techniques de pré-traitements utilisées dans quelques travaux de la littérature. Enfin, une conclusion est proposée en section 2.4.

2.2 Techniques de pré-traitement des données textuelles

Dans la littérature, les techniques de pré-traitements les plus communément utilisées sont : Tokenisation, Suppression des nombres, Conversion en minuscules, Suppression des mots vides, N-grammes, Racinisation et Lemmatisation. Ces différentes techniques sont détaillées dans la section suivante.

2.2.1 Tokenisation

Étant donnée une information textuelle, la tokenisation est le processus de décomposition d'un flux de texte en mots, phrases, symboles ou autres éléments significatifs appelés *'tokens'*. Dans notre contexte, la tokenisation consiste à explorer des mots d'une phrase. La liste des *tokens* devient une entrée pour un traitement ultérieur tel que l'analyse ou l'exploration de texte [161]. La tokenisation est utile à la fois en linguistique (où il s'agit d'une forme de segmentation de texte) et en informatique, où elle fait partie de l'analyse lexicale. Par exemple, un analyseur syntaxique effectue au préalable une tokenisation des documents. Certains problèmes peuvent subsister, comme le traitement des signes de ponctuation. D'autres caractères comme les crochets, les tirets, etc, nécessitent également un traitement spécifique. Un autre problème concerne les abréviations et les acronymes qui doivent être transformés en une forme standard [84].

2.2.1.1 Défis de la tokenisation

Les défis de la tokenisation dépendent du type de langue. Des langues tels que l'anglais et le français sont plus aisés a traiter car la plupart des mots sont séparés les uns des autres par des espaces. Des langues tels que le chinois et le thaï sont considérées comme non segmentées car les mots n'ont pas de frontières claires. La

tokenisation en phrases non segmentées nécessite des informations lexicales et morphologiques supplémentaires. La tokenisation est également impactée par le système d'écriture et la structure typographique du mots [132].

2.2.2 Suppression des nombres

Les *tokens* résultants de l'étape de tokenisation sont composés de caractères alphabétiques, alphanumériques ou numériques qui sont délimités par des caractères non alphanumériques. La suppression des nombres consiste à éliminer les *tokens* numériques du texte.

2.2.3 Conversion en minuscules

Une autre technique de pré-traitement largement utilisée en fouille de textes est la conversion en minuscules, qui sert à convertir tous les caractères majuscules en minuscules, afin d'éviter de considérer les mêmes mots écrits sous des formes différentes comme des mots différents, par exemple "WRITE", "Write" et "write". Ce regroupement sous un même *token* peut avoir un impact sur les méthodes numériques utilisées (calcule de fréquences de *tokens* par exemple)

Malgré sa propriété souhaitable de réduire la dispersion et la taille du vocabulaire, l'uniformisation peut avoir un impact négatif sur les performances des systèmes en augmentant l'ambiguïté [31]. Par exemple, la société "Apple" et le fruit "apple" seraient considérés comme des entités identiques.

2.2.4 Suppression des mots vides

Certains mots dans les documents reviennent très fréquemment mais sont essentiellement dénués de sens car ils sont utilisés pour joindre des mots dans une phrase, il s'agit de mots fonctionnels tels que les conjonctions, les articles, etc. Il est peu probable que ces mots courants - souvent appelés mots vides - véhiculent beaucoup d'informations. En raison de leur fréquence élevée d'occurrence, leur prise en compte par les algorithmes de fouille de texte présente un obstacle à l'analyse du contenu des documents [36].

Il existe plein de listes de mots vides, et la plupart des logiciels d'analyse de texte utilisent des listes de mots vides par défaut que les auteurs du logiciel ont tenté de créer pour fournir de bonnes performances dans la plupart des cas.

2.2.5 N-grammes

Les N-grammes représentent les groupes de N mots, correspondant à la séquence de N mots consécutifs dans un document. Dans un N-gramme, cette séquence de N mots est traitée comme une entité indivisible, et devient un pseudo-terme à part entière [2]. Un N-gramme de taille 1 est appelé «unigramme» ; la taille 2 est un «bigramme» ; la taille 3 est un «trigramme» ; etc. La figure 2.1 montre un exemple d'application de N-grammes sur un exemple de texte.

2.2.6 Racinisation

La technique de racinisation, ou ce que l'on appelle en anglais Stemming (le terme utilisé au cours de cette thèse) est le processus de réduction d'un mot à sa racine appelée 'stem', généralement en coupant la fin ou le début du mot, sur la base d'une liste de préfixes et suffixes que l'on trouve couramment dans un mot fléchi. Habituellement, les mots dérivés et leurs stems sont considérés comme sémantiquement similaires et pourraient être échangés, même si ce stem n'est pas en lui-même un stem valide, par exemple, les mots 'argue', argued, 'argues', 'arguing', et 'argus' partagent le même stem 'argu'. Les algorithmes de stemming dépendent du langage étudié.

Parmi les algorithmes les plus utilisés, nous citons l'algorithme de stemming de Martin Porter, *Porter Stemmer* [134] généralement défini par le processus de suppression des terminaisons morphologiques et inflexionnelles les plus courantes des mots. Jusqu'à présent, ce dernier est l'un des algorithmes les plus utilisés en raison de son extensibilité, de sa simplicité et de son adaptabilité. Porter a amélioré son algorithme en proposant Porter2, également appelé *Snowball Stemmer* [133], il est considéré comme un stemmer simple et rapide, avec un temps de calcul légèrement inférieur à celui du Porter. Le troisième algorithme utilisé dans cette étude est *Lancaster Stemmer* [129], qui est un algorithme simple et itératif qui supprime les fins d'un mot en un nombre indéfini d'étapes. C'est un algorithme de stemming très agressif dont les mots deviennent si court qu'ils sont à peine lisibles par un humain. Il fait partie des algorithmes les plus rapides et réduit énormément la taille de vecteur.

Erreurs en Stemming : Il y a principalement deux erreurs de Stemming [65] :

- Sur-stemming : lorsque deux mots avec des stems différentes sont issus de la même racine. Ceci est également connu comme un faux positif. Par exemple, les mots 'universal', 'university' et 'universe' sont converti en même stem 'univers' malgré que leurs significations modernes sont dans des domaines très différents, donc les traiter comme des synonymes réduira probablement la pertinence des résultats de recherche.
- Sous-stemming : lorsque deux mots qui devraient être issus de la même racine ne le sont pas. Ceci est également connu comme un faux négatif. Par exemple les mots 'alumnus', 'alumni' et 'alumnae' génèrent des stems différents, alors qu'ils ont des significations proches.

2.2.7 Lemmatisation

Dans de nombreuses langues, les mots apparaissent sous plusieurs formes fléchies. Par exemple, en anglais, le verbe 'to walk' peut apparaître comme 'walk', 'walked', 'walks' ou 'walking'. La forme de base, 'walk', que l'on pourrait rechercher dans un dictionnaire, est appelée le lemme du mot. La lemmatisation est étroitement liée au stemming [26].

Le stemming fait généralement référence à un processus heuristique grossier qui coupe les extrémités des mots. Cette coupe sans discernement peut réussir dans certaines occasions, mais pas toujours, et c'est pourquoi nous affirmons que cette approche présente certaines limites.

La lemmatisation se réfère généralement à faire les choses correctement avec l'utilisation d'un vocabulaire et une analyse morphologique des mots, visant normalement à supprimer uniquement les fins flexionnelles et à renvoyer la forme de base ou de dictionnaire d'un mot, qui est connue sous le nom de lemme. Un lemmatiseur ,nécessite un vocabulaire complet et une analyse morphologique pour lemmatiser correctement les mots. Cependant, les stemmers sont généralement plus faciles à implémenter et à exécuter plus rapidement [95]. Par exemple :

— Le mot 'better' a le mot 'good' comme lemme. Ce traitement nécessite une recherche dans un dictionnaire.

— Le mot 'walk' est la forme de base du mot 'walking', et par conséquent cela correspond à la fois au stemming et à la lemmatisation.

— Le mot 'meeting' peut être soit la forme de base d'un nom, soit la forme d'un verbe 'to meet' selon le contexte, par exemple, 'in our last meeting' ou 'We are meeting again tomorrow'. Contrairement au stemming, la lemmatisation tente de sélectionner le lemme correct en fonction du contexte.

2.2.8 Étiquetage morpho-syntaxique

L'étiquetage morpho-syntaxique ou ce que l'on appelle en anglais Part Of Speech (POS) tagging, également appelé balisage grammatical est l'une des tâches syntaxiques fondamentales du NLP, c'est le processus d'étiquetage d'un mot dans un texte (corpus) fondé sur le contexte. Généralement, les mots sont identifiés en tant que nom, verbe, article, adjectif, préposition, pronom, adverbe, conjonction et interjection [27].

Texte original : " And now for something completely different "

Texte étiqueté :

— 'And' / 'CC' : Conjonction de coordination

— 'now' / 'RB' : Adverbe

— 'for'/ 'IN' : Préposition ou conjonction subordonnée

— 'something' / 'NN' : Nom, singulier

— 'completely' / 'RB' : Adverbe

— 'different' / 'JJ' : Adjectif

La figure 2.1 montre plus d'exemples sur les différentes techniques de pré-traitement.

2.3 Pré-traitement des données textuelles pour la similarité sémantique

Le pré-traitement est l'une des étapes les plus importantes et cruciales du processus de la fouille de données. Il est alors important de connaître les techniques adaptées de pré-traitement qui améliorent la qualité des résultats de ce processus. À cette fin, le pré-traitement a déjà prouvé son impact positif dans la classification des textes [158], l'analyse des sentiments [52] et l'identification des auteurs [130]. Les résultats montrent que l'étape de pré-traitement est utile pour obtenir de meilleures performances, en tenant compte des particularités de chaque tâche pour choisir la combinaison de techniques qui peut convenir.

Exemple de texte : *In this study the size of population was chosen to be 10*

Suppression des nombres : *['In', 'be', 'chosen', 'of', 'population', 'size', 'study', 'the', 'this', 'to', 'was']*

Suppression des mots vides : *['10', 'In', 'chosen', 'population', 'size', 'study']*

Conversion en minuscules : *['10', 'be', 'chosen', 'in', 'of', 'population', 'size', 'study', 'the', 'this', 'to', 'was']*

3-grammes : *['10', 'In', 'In this', 'In this study', 'be', 'be 10', 'chosen', 'chosen to', 'chosen to be', 'of', 'of population', 'of population was', 'population', 'population was', 'population was chosen', 'size', 'size of', 'size of population', 'study', 'study the', 'study the size', 'the', 'the size', 'the size of', 'this', 'this study', 'this study the', 'to', 'to be', 'to be 10', 'was', 'was chosen', 'was chosen to']*

Racinisation : *['10', 'In', 'be', 'chosen', 'of', 'popul', 'size', 'studi', 'the', 'this', 'to', 'was']*

Lemmatisation : *['10', 'In', 'be', 'choose', 'of', 'population', 'size', 'study', 'the', 'this', 'to']*

FIGURE 2.1: Application de différentes techniques de pré-traitement sur un exemple de texte.

Concernant le domaine de la similarité sémantique, Mohamed et Oussalah [121] ont proposé une approche hybride qui aborde le problème de l'évaluation de la similarité sémantique phrase à phrase lorsque les phrases contiennent un ensemble d'entités nommées. Le système comprend quatre modules principaux : le pré-traitement du texte, la similarité sémantique des phrases, la conversion de PoS Word et la mesure de similarité WordNet. Le module de similarité sémantique de phrases représente le composant central du système. Les phrases pré-traitées sont nominalisés avant d'être introduites dans le sous-système central. Il faut noter que la phase de pré-traitement contient les étapes suivantes : la division en phrases, la tokenisation et la suppression des mots vides. Les résultats expérimentaux obtenus montrent que le système proposé a tendance à améliorer les baselines pour l'évaluation de similarité sémantique.

L'alignement de texte consiste à trouver des fragments de texte similaires entre deux documents donnés. Ceci a des applications dans la détection du plagiat, la détection de la réutilisation de texte, l'identification de l'auteur, l'aide à la création et la recherche d'informations, pour n'en citer que quelques-unes. Sanchez et al. [144] ont proposé une approche de la sous-tâche d'alignement de texte du défi de

détection de plagiat PAN 2014 [1], qui a abouti au système le plus performant. La méthode repose sur une mesure de similarité des phrases basée sur l'application d'une pondération de type tf-idf qui permet de considérer les mots vides sans augmenter le taux de faux positifs. Un algorithme récursif pour étendre les plages de phrases correspondantes aux passages de longueur maximale est introduit. Une nouvelle méthode de filtrage pour résoudre les cas de plagiat qui se chevauchent est également proposée. La phase de pré-traitement applique la division en phrases, la tokenisation, la conversion en minuscules et le stemming.

Amaral dans son article [12] a révisé la tâche d'identification des paraphrases (c'est-à-dire, étant donnée une paire de phrases, les classer comme étant des paraphrases ou non des paraphrases). Il a proposé d'aborder la tâche grâce à un apprentissage automatique supervisé, en formant des modèles de classification basés sur des ensembles d'arbres, qui utilisent des descripteurs qui correspondent principalement à des métriques de similarité de chaînes reposant sur des informations lexicales. Dans cette approche la phase de pré-traitement des phrases textuelles est composée de la tokenisation, le stemming et la lemmatisation.

2.4 Conclusion

Dans ce chapitre, nous avons présenté les différentes techniques de pré-traitement nécessaires pour comprendre et assimiler les solutions proposées et mises en oeuvre dans le cadre de cette thèse. Nous avons ensuite abordé les diverses techniques de pré-traitement suggérées dans la littérature et dédiées particulièrement à l'évaluation de similarité sémantique. Dans le chapitre suivant, nous nous intéressons aux outils et techniques de représentation de données.

1. https ://pan.webis.de/

Chapitre 3

Représentation vectorielle des données textuelles

Sommaire

3.1 Introduction

L'étape la plus fondamentale pour la majorité des tâches de NLP est de permettre aux ordinateurs de comprendre, d'analyser et d'extraire du sens du langage naturel humain, de manière intelligente et utile. En ce qui concerne les données textuelles, les ordinateurs sont plus aptes à gérer les valeurs numériques que le texte réel transmis sous forme de *tokens*. La tâche de transformation des mots en vecteurs numériques qui représentent leur signification s'appelle la représentation de texte. Cette étape, joue un rôle important dans le choix des caractéristiques de notre modèle / algorithme d'apprentissage automatique.

Les représentations vectorielles de texte peuvent être globalement classées en deux types : Représentations de texte discrètes et Représentations de texte distribuées.

La section 3.2 présente les méthodes de représentation vectorielle discrète. La section 3.3 décrit les méthodes de représentation vectorielle distribuée. Enfin, nous fournissons une conclusion en section 3.4.

3.2 Représentation vectorielle discrète

Une représentation discrète est une représentation où chaque mot est considéré comme unique et converti en numérique. Les représentations célèbres qui entrent dans cette catégorie sont : L'encodage One-Hot, sac de mots et l'encodage fréquence du terme / fréquence inverse de document [100].

3.2.1 Encodage One-Hot

L'encodage One-Hot est un encodage vectoriel booléen qui marque un index vectoriel particulier avec une valeur de $Vrai$ (1) si le *token* existe dans le document et de $Faux$ (0) dans le cas contraire. En d'autres termes, chaque élément d'un vecteur codé à One-Hot reflète soit la présence, soit l'absence du *token* dans le texte décrit [135].

Prenons cet exemple, où tous les mots, sans répétition, sont représentés :

1 : "Mary likes to see sun rise. John likes to play under bright sun."

2 : "The bright sun rise early."

3 : "See the moon rise".

La table 3.1 montre la représentation vectorielle de chaque phrase en utilisant L'encodage One-Hot, ça implique :

* Un vocabulaire de mots connus.

* Un état représentant l'absence ou la présence de chaque mot.

Malgré la simplicité de cette méthode d'encodage à mettre en œuvre, One-Hot ne fournit aucune information utile concernant l'ordre ou la fréquence des mots dans le document, ni aucune représentation des relations sémantiques qui peuvent exister entre les mots. En plus, la nature clairsemée des vecteurs générés conduit à une inefficacité en termes de coûts de stockage et de calcul.

Phrases \ Tokens	mary	likes	to	see	sun	rise	john	play	under	bright	the	early	moon
"Mary likes to see sun rise. John likes to play under bright sun".	1	1	1	1	1	1	1	1	1	1	0	0	0
"The bright sun rise early".	0	0	0	0	1	1	0	0	0	1	1	1	0
"See the moon rise".	0	0	0	1	0	1	0	0	0	0	1	0	1

TABLE 3.1: Encodage One-Hot

3.2.2 Sac de mots

L'encodage Sac de mots ou ce que l'on appelle en anglais Bag of Words (BoW) [143], est une représentation de texte qui décrit l'occurrence de mots dans un document, en impliquant :

* Un vocabulaire de mots connus.

* Une mesure de la présence de mots connus.

Si l'encodage One-Hot d'une phrase est composé de l'état de présence de chaque *token*, l'encodage BoW d'une phrase ou d'un document est simplement la somme de la représentation One-Hot de ses mots constitutifs. Prenons le même exemple précédent, la table 3.2 montre le résultat de représentation vectorielle en utilisant l'encodage BoW.

Phrases \ Tokens	mary	likes	to	see	sun	rise	john	play	under	bright	the	early	moon
"Mary likes to see sun rise. John likes to play under bright sun".	1	2	2	1	2	1	1	1	1	1	0	0	0
"The bright sun rise early".	0	0	0	0	1	1	0	0	0	1	1	1	0
"See the moon rise".	0	0	0	1	0	1	0	0	0	0	1	0	1

TABLE 3.2: Sac de mots

Malgré l'apport d'informations sur le nombre d'occurrence des mots dans le texte, BoW reste toujours une technique qui souffre de dispersion, de coût élevé et de manque d'information sémantique.

3.2.3 Fréquence du terme / Fréquence inverse de document

Les représentations BoW que nous avons présenté ne décrivent un document que de manière autonome, sans prendre en compte le contexte du corpus. Fréquence du

terme / Fréquence inverse de document ou ce que l'on appelle en anglais Term Frequency – Inverse Document Frequency (TF-IDF) [80, 58] est une variante avancée de représentation, qui prend en compte l'importance des mots en fonction de sa rareté dans le document (notion de discriminance). Cependant, au lieu de vecteurs avec des nombres d'occurrences discrets comme BoW, le vecteur généré contient une valeur continue selon l'équation suivante :

$$tfidf_{i,j} = tf_{i,j} * idf_i \tag{3.1}$$

$$tf_{i,j} = \frac{nombre\ d'occurrence\ de\ terme\ cible\ (i)}{nombre\ total\ de\ termes\ dans\ le\ document\ (j)} \tag{3.2}$$

$$idf_i = \log\left(\frac{nombre\ de\ documents\ dans\ le\ dataset}{nombre\ de\ documents\ contenant\ le\ terme\ cible\ (i)}\right) \tag{3.3}$$

En appliquant la représentation TF-IDF sur les mêmes exemples cités précédemment, la table 3.3 montre le résultat obtenu.

Tokens / Phrases	mary	likes	to	see	sun	rise	john	play	under	bright	the	early	moon
"Mary likes to see sun rise. John likes to play under bright sun".	0.477	0.620	0.620	0.176	0.228	0	0.477	0.477	0.477	0.176	0	0	0
"The bright sun rise early".	0	0	0	0	0.176	0	0	0	0	0.176	0.176	0.477	0
"See the moon rise".	0	0	0	0.176	0	0	0	0	0	0	0.176	0	0.477

TABLE 3.3: Pondération TF-IDF

Les représentations discrètes sont des représentations simples faciles à comprendre, à mettre en œuvre et à interpréter. Des pondérations tels que TF-IDF peuvent être utilisés pour filtrer les mots inhabituels et non pertinents, aidant facilement le modèle à former et à converger plus rapidement. Mais il souffre de différents inconvénients ; la représentation est proportionnelle à la taille du vocabulaire. Une taille de vocabulaire élevée peut entraîner des contraintes de mémoire. Elle ne prend pas en considération les cooccurrences entre les mots, elle suppose que tous les mots sont indépendants les uns des autres, aussi c'est une représentation qui conduit à des vecteurs très dispersés avec peu de valeurs non nulles, elles ne capture pas le contexte ou la sémantique du mot.

Malgré ses points faibles, les représentations discrètes sont largement utilisées dans les applications d'apprentissage automatique classiques et d'apprentissage en profondeur pour résoudre des cas d'utilisation complexes tels que la similarité de

documents [163], la classification des sentiments [7], la classification des spams [53] et la modélisation de sujets [55], etc.

3.3 Représentation vectorielle distribuée

Dans le but de remédier aux limites des représentations discrètes, de nouvelles techniques ont vu le jour, en particulier, les techniques de représentation distribuée. Elles se basent sur la recherche de toutes les occurrences d'un certain mot et de ses voisins sélectionnés via une fenêtre prédéfinie. Le vecteur du mot est ajusté en augmentant le nombre d'occurrences dans les indices de ses voisins.

Bien que la technique des vecteurs distributionnels capture des informations sur la similarité entre les mots, le stockage des vecteurs nécessite un espace mémoire très important, et l'exécution d'opérations sur tous les mots est complexe en termes de temps de calcul.

Depuis 2003 Bengio et al. [23] et afin de réduire la haute dimensionnalité des représentations de mots dans les contextes, ont fourni dans une série d'articles une nouvelle méthode de représentation, basée sur la sauvegarde des mêmes informations contextuelles dans un vecteur de faible dimension en "apprenant une représentation distribuée pour mot".

3.3.1 Plongement de mots

La représentation vectorielle distribuée des mots Word2Vec consiste à représenter les plongements de mots (en anglais Word embeddings). Il a été développé par Tomas Mikolov en 2013 [115]. Cette méthode se base sur des réseaux de neurones à deux couches et sert à apprendre les représentations vectorielles des mots composant un texte. Les mots qui partagent des contextes similaires sont alors représentés par des vecteurs numériques proches.

Les poids de la couche cachée sont le 'plongement' du mot et l'ajustement se fait via une fonction de perte (backpropagation normale). Cette architecture est similaire à celle d'un auto-encodeur, où on a une couche de codeur et une couche de décodeur et la partie centrale est la représentation compressée de l'entrée. Elle peut être utilisée pour la réduction de dimensionnalité.

La représentation Word2vec est créée à l'aide de deux algorithmes : l'un est le modèle sac de mots continu ou ce que l'on appelle en anglais Continious Bag of Words (CBoW) et le second est le modèle Skip-Gram.

— **CBoW** est un algorithme qui apprend à prédire le mot w_t en fonction de l'ensemble des mots de la fenêtre de contexte $\{w_{t-n}, ..., w_{t-1}, w_{t+1}, ..., w_{t+n}\}$. Durant l'étape d'apprentissage, la couche d'entrée reçoit un sac de mots binaires qui contient l'ensemble des mots de la fenêtre de contexte. Cette couche passe par une projection dans la couche cachée, ensuite par la couche de sortie pour prédire un mot.

  Une comparaison entre le mot central du contexte et la prédiction générée du réseau est réalisée. Durant l'apprentissage, les matrices de poids du modèle sont adaptées en se basant sur la rétro-propagation de l'erreur de prédiction.

— **Skip-Gram** sert à prédire le contexte $\{w_{t-n}, ..., w_{t-1}, w_{t+1}, ..., w_{t+n}\}$ pour un mot w_t donné. La couche d'entrée du Skip-Gram reçoit un vecteur représentant

le mot au centre du contexte. Elle est projetée dans la couche cachée du réseau pour produire une représentation dense, celle-ci est ensuite projetée à son tour dans la couche de sortie pour générer une prédiction. Cette prédiction est corrigée à base d'une comparaison par chaque mot contenu dans la fenêtre de contexte indépendamment, comme le montre la Figure 3.1.

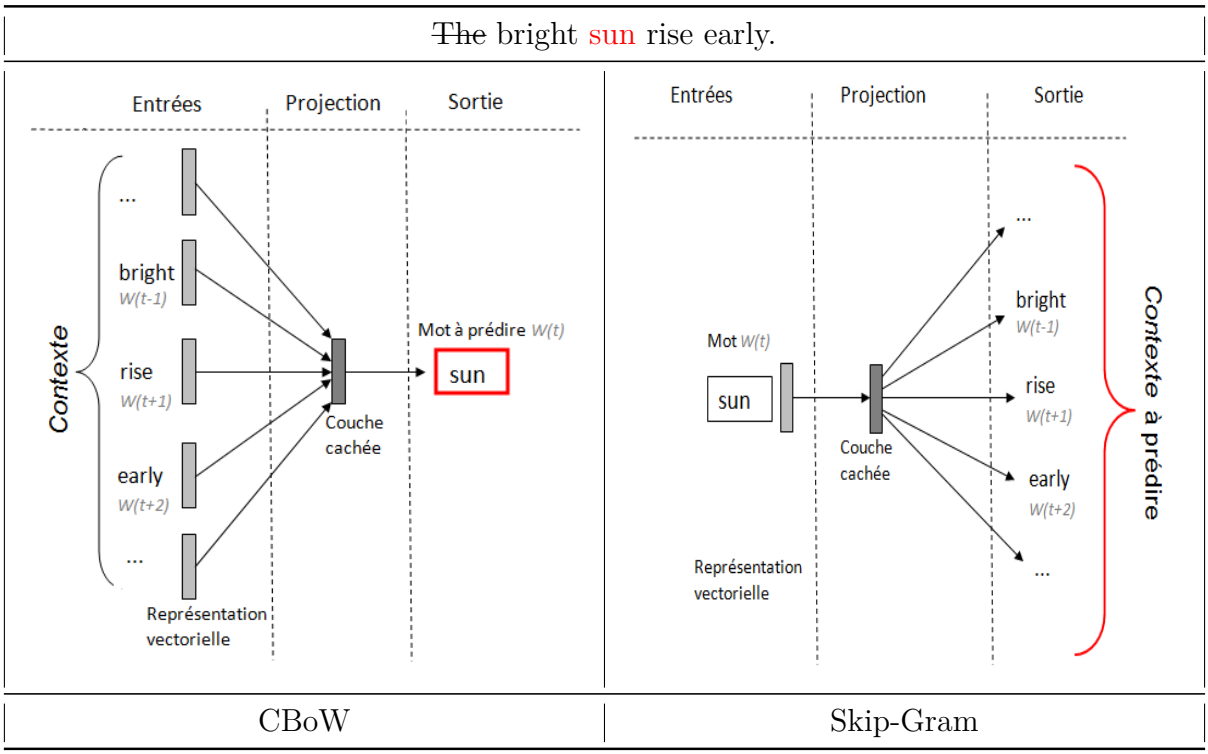

FIGURE 3.1: Architectures de CBoW et Skip-gram.

L'effet positif de cette technique obtenu en l'utilisant dans différentes tâches [152, 19, 154, 105, 157] a conduit les chercheurs à se demander comment tirer profit de cette approche dans le contexte des plongements de documents.

3.3.2 Plongement de documents

Au lieu de représenter un mot dans un vecteur, le plongement de documents (en anglais Document embedding) transforme un document complet en un espace vectoriel, basé sur les vecteurs de ses mots constitutifs. Ces dernières années, une large gamme de ces méthodes a été proposée [104, 92, 70, 34]. Toutes les méthodes présentées dans cette section sont inspirées par des techniques de plongement de mots, en particulier, word2vec [131].

3.3.2.1 Plongement de n-grammes

Mikolov et al. [116] ont étendu le modèle skip-gram de word2vec pour gérer des phrases courtes, les auteurs se concentrent sur des phrases de deux et trois mots, et les traitent comme *tokens* indivisibles lors de l'apprentissage du modèle word2vec. Naturellement, cela convient moins à l'apprentissage de phrases plus longues, car la

taille du vocabulaire explose lorsque la longueur de la phrase augmente et ne doit pas se généraliser aux phrases non vues ainsi qu'aux méthodes qui les suivent.

3.3.2.2 Calcul de la moyenne des plongements de mots

Il existe un moyen très intuitif de construire des plongements de documents à partir de plongements de mots significatifs : pour un document, la méthode effectue des calculs vectoriels sur tous les vecteurs correspondants aux mots du document pour les résumer en un seul vecteur dans le même espace de plongement ; deux de ces opérateurs de synthèse courants sont la moyenne et la somme.

Sur cette base, vous pouvez peut-être déjà imaginer que l'extension de l'architecture encodeur-décodeur de word2vec et de ses parents pour apprendre à combiner des vecteurs de mots dans des plongements de documents peut être intéressante ; les méthodes qui suivent celle-ci entrent dans cette catégorie.

Une deuxième possibilité consiste à utiliser un opérateur fixe (non apprenable) pour la représentation vectorielle, par ex. calcul de la moyenne, et apprendre les vecteurs de mots dans une couche précédente, en utilisant une cible d'apprentissage qui vise à produire des vecteurs de documents riches ; un exemple courant est l'utilisation d'une phrase pour prédire des phrases contextuelles. Ainsi, le principal avantage ici est que les vecteurs de mots sont optimisés pour la moyenne dans les représentations de documents.

Kenter et al [86], comme le montre la Figure 3.2, ont appliqué cette méthode, en utilisant un simple réseau de neurones sur une moyenne de vecteurs de mots. Les phrases environnantes sont alors apprises.

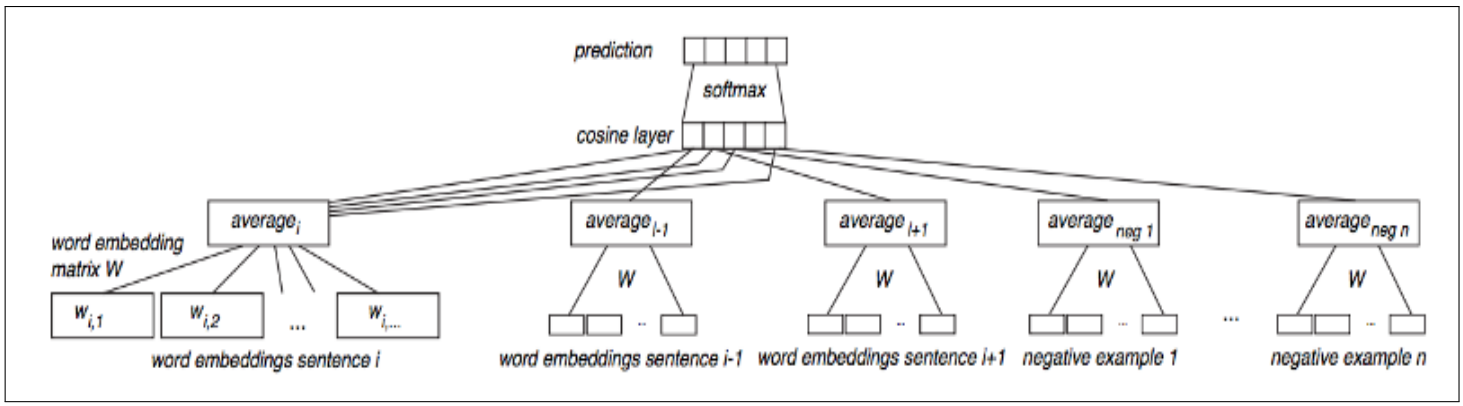

FIGURE 3.2: Architecture de réseau CBOW siamois de Kenter et al [86]

3.3.2.3 Vecteur de paragraphe / Doc2vec

Généralement appelé Doc2vec, vecteur de paragraphe ou ce que l'on appelle en anglais paragraph vector, a été développé par Le et Mikolov en 2014 [104] en tant qu'extension de Word2vec [115]. Cette méthode consiste à générer des vecteurs à partir du document dans son ensemble, au lieu de traiter seulement des mots individuels. Doc2vec peut être formé de manière totalement non supervisée à partir de texte brut. Cette méthode a un bon comportement dans le cas de la représentation de documents plus longs [103]. La représentation Doc2vec est produite en utilisant deux versions de vecteur de paragraphe : vecteur de paragraphe avec modèle de

mémoire distribuée ou ce que l'on appelle en anglais Paragraph Vector - Distributed Memory (PV-DM) et vecteur de paragraphe avec sac de mots distribué ou ce que l'on appelle en anglais Paragraph Vector - Distributed Bag of Words (PV-DBoW).

— **PV-DM**, en plus du codeur-décodeur standard qui convertit chaque mot en un seul vecteur, le modèle génère également un vecteur mémoire qui représente le sujet de chaque paragraphe. En sortie, un mot doit être prédit à partir de son contexte via un processus de moyenne et de concaténation.

— **PV-DBoW**, sert à prédire un mot de contexte unique en ignorant les mots de contexte dans l'entrée et en forçant le modèle à être basé uniquement sur le vecteur de paragraphe. Le fait que les vecteurs de paragraphe ne soient pas appris avec les vecteurs de mots peut améliorer considérablement les performances du modèle en termes d'exécution et de mémoire.

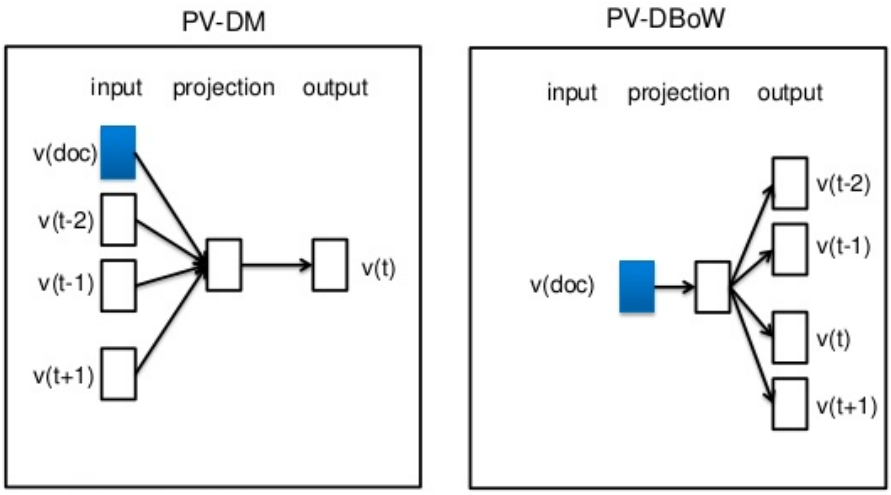

FIGURE 3.3: Architectures PV-DM et PV-DBOW.

Cette approche a surpassé la représentation de document précédemment décrite, sur diverses tâches de compréhension de texte [40]. Malgré ses avantages de permettre la généralisation à des documents plus longs, d'apprendre à partir de données non étiquetées, de prendre en compte l'ordre des mots, etc., l'approche Doc2vec souffre encore de deux limites [34] : le nombre de paramètres augmente avec la taille du corpus d'apprentissage, qui peut facilement atteindre des milliards ; et il est coûteux de générer des représentations vectorielles pour des documents non vus au moment du test. Cela a conduit à proposer une architecture de modèle améliorée, appelée Vecteur de document à travers la corruption.

3.3.2.4 Vecteur de document à travers la corruption / Doc2vecC

Avec la même architecture Doc2vec, la méthode Doc2vecC [34] est composée d'une couche d'entrée, d'une couche de projection et d'une couche de sortie.

Deux contributions ont amélioré Doc2vecC par rapport à Doc2vec. Premièrement, Doc2VecC représente chaque document comme une moyenne des 'plongements' de mots échantillonnés aléatoirement, contrairement à Doc2vec qui apprend directement un vecteur unique pour chaque document. Deuxièmement, le concept de 'corruption' consiste à la suppression aléatoire d'une partie des mots du document

d'origine, et le vecteur de document est généré en faisant la moyenne des vecteurs des mots restants.

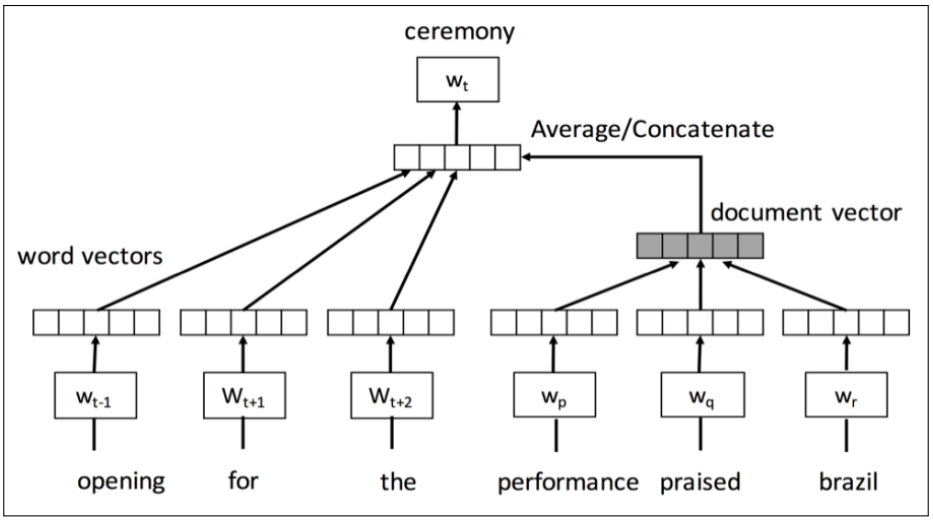

FIGURE 3.4: Architecture de Doc2vecC [34].

Comme le montre la figure 3.4, les vecteurs de mots voisins, «openning», «for», «the», fournissent un contexte local, tandis que la représentation vectorielle du document complet, représentée en gris, sert du contexte. Chaque document est représenté comme une moyenne des vecteurs de mots pris au hasard dans le document. Dans la figure 3.4 les vecteurs de mots prises sont les mots «performance» à la position p, «praised» à la position q et «brazil» à la position r.

Selon [34], l'approche Doc2vecC a surpassé l'approche originale Doc2vec dans plusieurs tâches. Malgré cela, le plongement de documents est un sujet encore activement exploré pour ses diverses caractéristiques,

3.3.3 Applications de plongement de documents

La capacité de mapper des documents sur des représentations vectorielles informatives a un large éventail d'applications. Ce qui suit n'est qu'une liste non exhaustive.

Le et Mikolov [104] ont démontré les capacités de leur méthode de vecteurs de paragraphe sur plusieurs tâches de classification de texte et d'analyse des sentiments, tandis que Dai et al [40] l'ont examinée dans le contexte de tâches de similarité de documents et Lau et Baldwin [103] contre une tâche de duplication de question de forum et la compétition SemEval de similarité textuelle sémantique STS.

Kiros et al. [92] a démontré l'utilisation de leurs vecteurs de saut de pensée pour la relation sémantique, la détection de paraphrases, le classement image-phrase, la classification par type de question et quatre ensembles de données sur le sentiment et la subjectivité. Broere [29] a utilisé ces méthodes pour prédire les balises POS et les relations de dépendance.

Chen et al. [35] ont montré que leur ensemble de plongements de phrases formés sur des textes biomédicaux, fonctionnait bien sur les tâches de similarité de paires de phrases.

Enfin, le modèle de similarité sémantique profonde a été utilisé par divers auteurs pour la recherche d'informations et le classement de la recherche sur le Web, la sélec-

tion / pertinence des annonces, la recherche d'entités contextuelles et les tâches d'intérêt, la réponse aux questions, l'inférence de connaissances, le sous-titrage d'images et les tâches de traduction automatique.

3.4 Conclusion

Ce chapitre introduit les concepts fondamentaux de la représentation vectorielle des données ; nous y avons en particulier défini son importance et ses différentes stratégies et méthodes discrètes et distribuées. Dans le chapitre suivant, nous nous intéressons à la sélection des caractéristiques en détaillant les méthodes et algorithmes.

Chapitre 4

Sélection des caractéristiques fondée sur les métaheuristiques

Sommaire

4.1 Introduction

La sélection des caractéristiques est également appelée sélection de variables ou sélection d'attributs. Cette tâche correspond au processus de réduction de la dimensionnalité ou du nombre de variables d'entrée lors du développement d'un modèle prédictif. Il est souhaitable de réduire le nombre de variables d'entrée à la fois pour réduire le coût de calcul de la modélisation et, dans certains cas, pour améliorer les performances du modèle

La détermination des caractéristiques à inclure dans un modèle devient l'une des questions les plus critiques, car les données sont de plus en plus dimensionnelles [97] avec des exemples d'applications données ci-dessous :

— En affaires, les entreprises sont désormais plus compétentes pour stocker et accéder à de grandes quantités d'informations sur leurs clients et leurs produits. De grandes bases de données sont souvent exploitées pour découvrir des relations cruciales [109].

— Dans la recherche pharmaceutique, les chimistes peuvent calculer des milliers de caractéristiques à l'aide des méthodologies spécialisées pour décrire numériquement divers aspects des molécules. Ces caractéristiques peuvent être discrètes ou continues et peuvent facilement se chiffrer en dizaines de milliers [97].

— En biologie, un vaste éventail de caractéristiques biologiques peut être mesuré en même temps sur un échantillon de matériel biologique tel que le sang. Les microréseaux de profilage d'expression d'ARN peuvent mesurer des milliers de séquences d'ARN à la fois. En outre, les puces à ADN et les technologies de séquençage peuvent déterminer de manière exhaustive la composition génétique d'un échantillon, produisant une multitude de caractéristiques numériques. Ces technologies ont rapidement évolué au fil du temps, offrant des quantités d'informations de plus en plus importantes [109].

D'un point de vue pratique, un modèle avec moins de caractéristiques peut être plus interprétable et moins coûteux, surtout s'il y a un coût pour mesurer ces caractéristiques. Statistiquement, il est souvent plus intéressant d'estimer moins de paramètres. De plus, certains modèles peuvent être affectés négativement par des caractéristiques non informatives.

L'importance de sélection des caractéristiques est présentée en section 4.2. La section 4.3 donne une description des méthodes utilisées pour la sélection des caractéristiques. La section 4.4 représente quelque métaheuristiques utilisées pour la sélection des caractéristiques. Enfin, nous fournissons une conclusion en section 4.5.

4.2 Importance de sélection des caractéristiques

Dans cette section, nous allons présenter les avantages et l'intérêt derrière l'utilisation de la sélection des caractéristiques dans un modèle d'apprentissage automatiques [172] :

— En utilisant la sélection de caractéristiques, nous pouvons supprimer les caractéristiques non pertinentes qui n'affecteraient pas ou ne modifieraient pas la sortie de notre modèle. Par exemple, si nous essayons de prédire le prix

d'une maison en Espagne, en utilisant des variables qui incluent les conditions météorologiques en Chine, ces variables ne seront probablement pas très utiles !

— Ces types de caractéristiques non pertinentes peuvent en fait diminuer les performances du modèle en introduisant du bruit.

— Moins de caractéristiques signifie généralement des modèles d'entraînement plus rapides : pour les modèles paramétriques comme la régression linéaire ou logistique, cela signifie qu'il y a moins de poids à calculer, et pour les modèles non paramétriques comme les arbres de décision de forêt aléatoire, cela signifie qu'il y a moins de caractéristiques à évaluer à chaque division.

— Lors de la mise en production de modèles, moins de caractéristiques signifie moins de travail pour l'équipe qui crée l'application qui utilisera le modèle. En utilisant la sélection des caractéristiques, nous pouvons réduire le temps d'intégration de l'application.

— Lorsque nous conservons les caractéristiques les plus importantes, en supprimant celles que nos méthodes de sélection de caractéristiques jugent non pertinents, notre modèle devient plus simple et plus facile à comprendre. Par exemple un modèle avec 25 caractéristiques est beaucoup plus simple qu'un modèle avec 200 caractéristiques.

— Une fois l'application terminée et utilisée périodiquement, un modèle avec moins de caractéristiques est beaucoup plus facile à déboguer en cas de comportement anormal qu'un modèle avec beaucoup de caractéristiques [172].

4.3 Méthodes de sélection de caractéristiques

Une distinction importante à faire dans la sélection des caractéristiques est celle des méthodes supervisées et non supervisées. Lorsque le résultat est ignoré lors de l'élimination des caractéristiques, la technique n'est pas supervisée. Pour les méthodes supervisées, les caractéristiques sont spécifiquement sélectionnées dans le but d'augmenter la précision ou de trouver un sous-ensemble de caractéristiques afin de réduire la complexité du modèle. Ici, le résultat est généralement utilisé pour quantifier l'importance des caractéristiques. Les problèmes liés à chaque type de sélection de caractéristiques sont très différents et la littérature sur ce sujet est vaste [97].

La plupart des approches pour réduire le nombre de caractéristiques peuvent être classées en trois catégories principales détaillées dans les sous-sections suivantes.

4.3.1 Les méthodes d'emballage

Les méthodes d'emballage ou appelées aussi méthodes wrapper (voir la figure 4.1), évaluent plusieurs modèles à l'aide de procédures qui ajoutent et / ou suppriment des caractéristiques pour trouver la combinaison optimale qui maximise les performances du modèle. Essentiellement, les méthodes d'emballage sont des algorithmes de recherche qui traitent les caractéristiques comme des entrées et utilisent les performances du modèle comme sortie à optimiser [122].

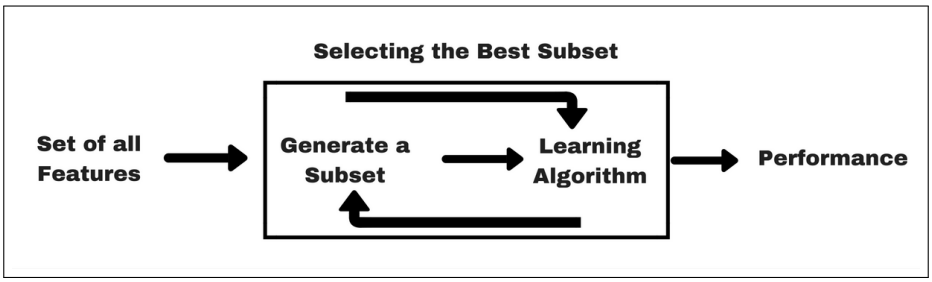

FIGURE 4.1: Sélection des caractéristiques par méthodes d'emballage

4.3.2 Les méthodes de filtrage

Les méthodes de filtrage ou appelées aussi méthodes Filter (voir la figure 4.2), évaluent la pertinence des caractéristiques en dehors des modèles prédictifs et modélisent par la suite uniquement les caractéristiques qui satisfont à certains critères [97]. Par exemple, pour les problèmes de classification, chaque caractéristique pourrait être évaluée individuellement pour vérifier s'il existe une relation plausible entre lui et les classes observées. Seules les caractéristiques ayant des relations importantes seraient alors intégrées dans un modèle de classification.

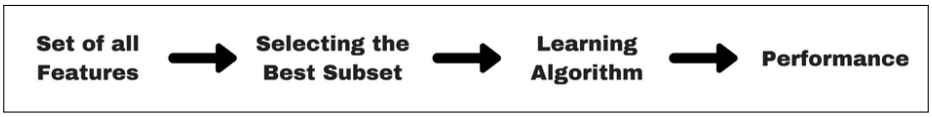

FIGURE 4.2: Sélection des caractéristiques par méthodes de filtrage

4.3.3 Les méthodes intégrées

Dans les méthodes intégrées (voir la figure 4.3), le processus de sélection des caractéristiques fait partie intégrante du modèle de classification [97].

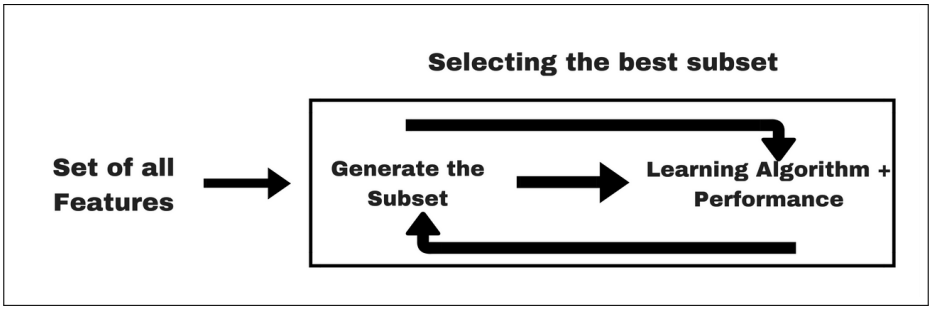

FIGURE 4.3: Sélection des caractéristiques par méthodes intégrées

4.3.3.1 Différence entre les méthodes d'emballage et de filtrage

— Les méthodes de filtrage mesurent la pertinence des caractéristiques par leur corrélation avec la variable dépendante, tandis que les méthodes d'emballage mesurent l'utilité d'un sous-ensemble de caractéristiques en entraînant réellement un modèle dessus.

— Les méthodes de filtrage sont beaucoup plus rapides que les méthodes d'emballage car elles n'impliquent pas l'apprentissage des modèles. D'un autre côté, les méthodes d'emballage sont également très coûteuses en calcul.

— Les méthodes de filtrage utilisent des méthodes statistiques pour l'évaluation d'un sous-ensemble de caractéristiques, tandis que les méthodes d'emballage utilisent la validation croisée.

— Les méthodes de filtrage peuvent ne pas trouver le meilleur sous-ensemble de caractéristiques dans de nombreuses occasions, mais les méthodes d'emballage peuvent toujours fournir le meilleur sous-ensemble de caractéristiques.

— L'utilisation du sous-ensemble de caractéristiques des méthodes d'emballage rend le modèle plus sujet au sur-apprentissage [1] par rapport à l'utilisation d'un sous-ensemble de caractéristiques des méthodes de filtrage [85].

4.4 Sélection des caractéristiques par les Métaheuristiques

Les algorithmes appelés métaheuristiques sont utilisés dans une variété de domaines [22], tels que l'administration de gestion, les statistiques et tous les domaines de l'ingénierie et de l'informatique. Ils sont capables de créer des moyens dynamiques adaptés à la nature du problème à traiter et identifier la solution la plus appropriée parmi l'éventail de solutions possibles à ce problème avant d'améliorer la valeur de cette solution au maximum. Les métaheuristiques prennent en compte les informations collectées lors de la recherche pour piloter le processus de recherche. Ils génèrent de nouvelles solutions en combinant une ou plusieurs bonnes solutions. Les métaheuristiques sont généralement des méthodes incomplètes ; ils ne garantissent pas de trouver la meilleure solution au niveau universel. Ils trouvent souvent des résultats d'approximation [8].

Lors de la conception d'algorithmes métaheuristiques, l'exploration et l'exploitation doivent être prises en compte [11] :

— La phase d'exploration est la possibilité de déplacer vers plusieurs régions de l'espace de recherche afin d'évaluer des solutions candidates qui ne sont pas voisines de la solution actuelle. Les algorithmes métaheuristiques basés sur la population sont orientés vers l'exploration.

— La phase d'exploitation est lorsqu'une recherche est effectuée dans le voisinage de la solution actuelle. Elle peut être mis en œuvre comme une recherche locale. Les algorithmes métaheuristiques à base unique sont orientés vers l'exploitation.

4.4.1 Algorithmes métaheuristiques à base unique

Les algorithmes métaheuristiques à base unique sont basés sur la création d'une première solution du problème puis ils s'en éloignent vers les régions voisines. Ils sont appelés «algorithmes de trajectoire» car ils suivent un chemin spécifique dans le processus de recherche, en commençant par la première solution. Dans cette méthode, chaque étape est exécutée uniquement si la solution de résultat est meilleure

1. https ://www.lexico.com/definition/overfitting

que la solution actuelle et le processus se poursuit jusqu'à ce qu'un minimum local soit obtenu [28].

4.4.1.1 Recuit simulé

Le recuit simulé ou ce que l'on appelle en anglais Simulated Annealing (SA) a été largement utilisé dans les problèmes d'optimisation. L'idée de cet algorithme vient du procédé acier [114], qui simule le refroidissement d'un matériau dans un bain de chaleur. Cet algorithme s'intéresse à l'étude des propriétés de la structure du matériau après refroidissement, car ces propriétés sont affectées par la vitesse de refroidissement. Si le processus de refroidissement est lent, de grosses perles de cristal se formeront, mais si le contraire se produit, les cristaux auront des défauts [69].

SA a prouvé ses performances dans la sélection de caractéristiques [111, 72, 1], ainsi que dans diverses applications de NLP, à savoir la désambiguïsation lexicale [37], l'apprentissage non supervisé des étiqueteurs POS, l'induction de grammaire [151], la classification basée sur un dictionnaire [93], la génération de paraphrases non supervisée [108], etc.

4.4.1.2 Recherche tabou

La recherche tabou ou ce que l'on appelle en anglais Tabu Search (TS) est une métaheuristique d'optimisation présentée par Fred W. Glover [60] en 1986.

TS utilise une procédure de recherche locale ou de voisinage pour passer de manière itérative d'une solution potentielle x à une solution améliorée $x\prime$ au voisinage de x, jusqu'à ce qu'un critère d'arrêt ait été satisfait (généralement, une limite de tentative ou un seuil de score). Les procédures de recherche locales sont souvent bloquées dans les zones à faible score ou dans les zones où les scores plafonnent. Afin d'éviter ces écueils et d'explorer des régions de l'espace de recherche qui seraient laissées inexplorées par d'autres procédures de recherche locales, TS explore attentivement le voisinage de chaque solution au fur et à mesure de la progression de la recherche. Les solutions admises dans le nouveau voisinage, $N^*(x)$, sont déterminées par l'utilisation de structures de mémoire. En utilisant ces structures mémoires, la recherche progresse en passant de manière itérative de la solution courante x à une solution améliorée $x\prime$ dans $N^*(x)$ [63]. TS présente plusieurs similarités avec SA, car les deux impliquent de possibles descentes de collines. En fait, SA pourrait être considéré comme une forme spéciale de TS, où nous utilisons la «tenure graduée», c'est-à-dire qu'un mouvement devient tabou avec une probabilité spécifiée [76].

TS a prouvé ses performances dans la sélection de caractéristiques [169, 127, 45], ainsi que dans diverses applications de NLP, à savoir la classification basée sur un dictionnaire [93], la catégorisation des textes [32], l'analyse des sentiments [61], etc.

4.4.2 Algorithmes métaheuristiques à base population

Les métaheuristiques à base population traitent un ensemble de solutions plutôt qu'une solution unique. Dans un premier temps, un ensemble de solutions est initialisé, puis un autre ensemble de solutions (population) est généré. Les étapes de recherche sont arrêtées après avoir atteint un certain critère [59]. Dans ces al-

gorithmes, il convient de noter que l'atteinte de la solution optimale dépend de la manière dont la population est manipulée.

4.4.2.1 Algorithme génétique

Cette approche d'optimisation est basé sur les principes évolutifs de la biologie des populations et s'est avéré efficace pour trouver des solutions optimales de fonctions complexes et multivariées. Plus précisément, un algorithme génétique ou ce que l'on appelle en anglais Genetic Algorithm (GA) est construit pour imiter le processus évolutif en permettant à la population actuelle de solutions de se reproduire, générant des enfants qui rivalisent pour survivre. Les survivants les plus aptes sont alors autorisés à se reproduire, créant ainsi la prochaine génération d'enfants. Au fil du temps, les générations convergent vers un plateau de fitness [71] et une solution optimale peut être choisie.

Le problème de la sélection des caractéristiques est intrinsèquement un problème d'optimisation complexe, où nous recherchons la combinaison de caractéristiques qui fournit une prédiction optimale de la réponse. Pour utiliser les GAs à cette fin, nous devons définir le problème de sélection des caractéristiques. Dans ce contexte, nous considérons les chromosomes, constitués de gènes et évalués en fonction de leur comportement. Pour créer la prochaine génération de progéniture, deux chromosomes se reproduisent grâce au processus de croisement et de mutation. Les GAs se sont révélées être des approches efficaces de sélection de caractéristiques dans différents domaines [166, 81, 128, 17], ainsi que dans diverses applications de NLP [87, 73, 16].

Algorithm 1 : GA pour la sélection des caractéristiques.

Début
    Définir les critères d'arrêt, le nombre d'enfants pour chaque génération (GenSize) et la probabilité de mutation (pm)
    Générer un ensemble aléatoire initial de m chromosomes binaires, chacun de longueur p
    Répéter
        Pour chaque chromosome **Faire**
            Ajuster et entraîner un modèle et calculer la forme physique de chaque chromosome
        Fin Pour
        Pour reproduction k = 1 . . . GenSize/2 **Faire**
            Sélectionner deux chromosomes en fonction du critère de fitness
            Crossover : sélectionner au hasard un loci et échanger chacun des gènes du chromosome au-delà des locus
            Mutation : changer aléatoirement les valeurs binaires de chaque gène dans chaque nouveau chromosome enfant avec probabilité, pm
        Fin Pour
    Jusqu'à les critères d'arrêt sont remplis
Fin

Dans le contexte de la sélection de caractéristiques, le chromosome est un vecteur binaire qui a la même longueur que le nombre de caractéristiques dans l'ensemble de données. Chaque entrée binaire du chromosome, ou gène, représente la présence ou

l'absence de chaque caractéristique dans les données. L'aptitude du chromosome est déterminée par le modèle en utilisant les caractéristiques indiquées par le vecteur binaire. Les GAs sont donc chargés de trouver des solutions optimales parmi les 2n combinaisons possibles d'ensembles de caractéristiques.

Pour commencer, les GAs sont souvent lancées avec une sélection aléatoire de chromosomes parmi la population de tous les chromosomes possibles. L'aptitude de chaque chromosome est calculée, ce qui détermine la probabilité de sélection du chromosome pour le processus de reproduction. Deux chromosomes de la population actuelle sont ensuite sélectionnés sur la base du critère de fitness et sont autorisés à se reproduire. Dans la phase de reproduction, les deux chromosomes parents sont séparés à une position aléatoire (également appelée locus) et la tête d'un chromosome est combinée avec la queue de l'autre chromosome et vice versa. Après le croisement, les entrées individuelles des nouveaux chromosomes peuvent être sélectionnées au hasard pour une mutation dans laquelle la valeur binaire actuelle est remplacée par l'autre valeur. L'algorithme 1 répertorie ces étapes.

La phase de croisement conduit les générations suivantes vers des optimums dans des sous-espaces de matériel génétique similaire. En d'autres termes, le sous-espace de recherche sera réduit à l'espace défini par les chromosomes les plus adaptés. Cela signifie que l'algorithme pourrait être piégé dans un optimum local. Dans le contexte de la sélection de caractéristiques, cela signifie que les caractéristiques sélectionnées peuvent produire un modèle optimal, mais que d'autres sous-ensembles de caractéristiques plus optimales peuvent exister [120].

La phase de mutation permet à l'algorithme d'échapper aux optimum locaux en perturbant aléatoirement le matériel génétique. Habituellement, la probabilité de mutation est maintenue faible (par exemple, pm $< 0{,}05$). Cependant, si le l'utilisation est préoccupé par les optimums locaux, alors la probabilité de mutation peut être augmentée. L'effet de l'augmentation de la probabilité de mutation est un ralentissement de la convergence vers une solution optimale [162].

4.4.2.2 Colonies de fourmis

L'optimisation des colonies de fourmis ou ce que l'on appelle en anglais Ant Colony Optimization (ACO) est une métaheuristique basée sur la population qui peut être utilisée pour trouver des solutions approximatives à des problèmes d'optimisation difficiles [47].

Dans ACO, un ensemble d'agents logiciels appelés fourmis artificielles recherche de bonnes solutions à un problème d'optimisation donné. Pour appliquer ACO, le problème d'optimisation se transforme en problème de recherche du meilleur chemin sur un graphe pondéré. Les fourmis artificielles (appelées fourmis) construisent progressivement des solutions en se déplaçant sur le graphique. Le processus de construction de la solution stochastique est biaisé par un modèle de phéromone, c'est-à-dire un ensemble de paramètres associés aux composants du graphe (nœuds ou arêtes) dont les valeurs sont modifiées à l'exécution par les fourmis [50].

Le moyen le plus simple de comprendre le fonctionnement de l'optimisation des colonies de fourmis est d'utiliser un exemple. Nous considérons son application au problème du voyageur de commerce ou ce que l'on appelle en anglais Travelling Salesman Problem (TSP) [48]. Dans le TSP, un ensemble d'emplacements (par exemple des villes) et les distances entre eux sont indiqués. Le problème consiste à trouver un circuit fermé de durée minimale qui visite chaque ville une et une seule fois.

Pour appliquer ACO au TSP, on considère le graphe défini en associant l'ensemble des villes à l'ensemble des sommets du graphe. Ce graphe est appelé graphe de construction. Comme dans le TSP, il est possible de se déplacer d'une ville donnée à n'importe quelle autre ville, le graphe de construction est entièrement connecté et le nombre de sommets est égal au nombre de villes. Nous définissons les longueurs des arêtes entre les sommets pour être proportionnelles aux distances entre les villes représentées par ces sommets et nous associons des valeurs de phéromone et des valeurs heuristiques aux arêtes du graphique. Les valeurs des phéromones sont modifiées à l'exécution et représentent l'expérience cumulée de la colonie de fourmis, tandis que les valeurs heuristiques sont des valeurs dépendant du problème qui, dans le cas du TSP, sont définies pour être l'inverse des longueurs des arêtes.

Les fourmis construisent les solutions comme suit. Chaque fourmi part d'une ville choisie au hasard (sommet du graphe de construction). Ensuite, à chaque étape de construction, elle se déplace le long des bords du graphique. Chaque fourmi garde une mémoire de son chemin, et dans les étapes suivantes, elle choisit parmi les arêtes qui ne mènent pas aux sommets qu'elle a déjà visités. Une fourmi a construit une solution une fois qu'elle a visité tous les sommets du graphe. À chaque étape de construction, une fourmi choisit de manière probabiliste l'arête à suivre parmi celles qui mènent à des sommets non encore visités. La règle probabiliste est biaisée par les valeurs de phéromone et les informations heuristiques : plus la phéromone et la valeur heuristique associées à un bord sont élevées, plus la probabilité qu'une fourmi choisisse ce bord particulier est élevée. Une fois que toutes les fourmis ont terminé leur tournée, la phéromone sur les bords est mise à jour. Chacune des valeurs de phéromone est initialement diminuée d'un certain pourcentage. Chaque arête reçoit alors une quantité de phéromone supplémentaire proportionnelle à la qualité des solutions auxquelles elle appartient (il y a une solution par fourmi). Cette procédure est appliquée à plusieurs reprises jusqu'à ce qu'un critère de terminaison soit satisfait.

ACO a prouvé ses performances dans la sélection de caractéristiques [3, 82], ainsi que dans diverses applications de NLP [9, 6].

4.5 Conclusion

Dans ce chapitre, nous avons introduit les concepts fondamentaux de la sélection des caractéristiques ; nous y avons en particulier défini son importance et ses différentes stratégies et méthodes d'emballage, de filtrage et intégrées. Nous avons décrit par la suite les métaheuristiques appliquées pour la sélection des caractéristiques, en prenant le recuit simulé et l'algorithme génétique comme exemples. Dans la deuxième partie du présent manuscrit, nous allons présenter nos contributions.

Deuxième partie

Contributions

Chapitre 5

Mise en correspondance de données textuelles hétérogènes à partir de Thésaurus

Sommaire

5.1 Introduction

Pour traiter les masses de données issues du Web disponibles, la problématique de recherche du Big Data est classiquement mise en avant avec les 3V qui la caractérisent : volume, variété et vélocité. Même si une distinction est établie entre la véracité (qualité) et la variété (hétérogénéité) des données, l'imbrication de ces deux concepts doit être prise en compte. En effet, pour avoir une connaissance exhaustive d'un sujet donné, il est nécessaire de traiter et de mettre en relation les données hétérogènes et de qualité différente. Ceci améliore indéniablement ce que nous pouvons appeler la qualité des connaissances traitées en prenant en compte différents points de vue. En effet, une problématique tout à fait ouverte consiste à analyser des situations en considérant les multiples points de vue à travers les dires d'acteurs et d'experts. Par exemple, lorsque l'on parle de changement climatique (jeux de données traité dans ce manuscrite) plusieurs positions peuvent être mises en relief sur des supports différents (articles de presse, tweets, articles scientifiques, etc.). La qualité des connaissances est donc fortement liée à la diversité des points de vue abordés sur un sujet donné. Dans ce contexte, nous nous intéressons à la manière de mettre en correspondance des données textuelles hétérogènes qui sont, par nature, de qualité diverse (formats, contenus, styles linguistiques, etc.).

Ce chapitre de thèse est organisé comme suit. Les motivations de cette contribution sont présentées en section 5.2. Dans la section 5.3, nous présentons les éléments de contributions utilisés dans nos travaux. L'approche proposée pour la mise en correspondance des documents textuels est présentée en section 5.4. Une description des expérimentations et des résultats obtenus est présentée en section 5.5. Enfin, nous terminerons ce chapitre par une conclusion et quelques perspectives qui sont présentées en section 5.6.

5.2 Motivation

Plusieurs projets de recherche s'intéressent à la similarité sémantique entre des extraits de textes, mais la plupart d'entre eux s'appuient sur des textes ayant un même niveau linguistique et stylistique [112, 54]. Le développement d'une approche efficace qui permet de proposer une similarité sémantique entre les textes hétérogènes représente alors une problématique éminemment difficile. Il existe un certain nombre de travaux de la littérature liés à l'estimation de similarité sémantique, dont beaucoup sont fondés sur l'utilisation de thésaurus. Par exemple, Buscaldi et al. [30] proposent un processus de comparaison de n-grammes sur la base d'une mesure de similarité conceptuelle utilisant WordNet [118]. Ils ont aussi appliqué une démarche similaire pour calculer la similarité sémantique de fragments textuels. Les textes peuvent être écrits selon des styles très différents, par exemple, les tweets sont beaucoup plus difficiles à analyser linguistiquement. A contrario, les articles scientifiques ont une écriture plus standardisée permettant l'extraction d'information de manière plus aisée. Mais ce type de textes possède un vocabulaire de spécialité souvent plus complexe [21]. Dans ce chapitre, nous proposons l'approche MIGHT (a text mining process for MappInG HeTeroeneous documents) pour mesurer la similarité sémantique entre les textes hétérogènes et de qualité différente. Notre approche utilise un système extraction que nous avons implanté et qui s'appuie sur le thésaurus multilingue AGROVOC. Nous avons alors combiné les informations extraites pour

calculer la similarité entre les représentations textuelles.

5.3 Éléments de contribution

Dans cette section, nous allons présenter les éléments de contribution utilisés dans nos travaux liés a l'approche MIGHT proposée.

5.3.1 Thésaurus

Les thésaurus peuvent désigner un certain nombre de ressources linguistiques. Nous travaillons à partir d'une définition d'un thésaurus : 'une ressource dans laquelle des mots ayant des significations similaires sont regroupés' [91]. Plusieurs types de thésaurus peuvent être utilisés :

— **Roget** Roget, Macquarie et autres, ont été produit, sous forme de livres, pour aider les écrivains dans la sélection des mots [43].

— **WordNet** WordNet et EuroWordNet est une base de données lexicale produite sur les principes psycholinguistiques. Elle a été développé à l'Université de Princeton et maintenant, est disponible gratuitement. Elle a été très largement utilisé dans la recherche en ingénierie linguistique [117].

— **IR-manual** Thésaurus produit manuellement pour une utilisation dans les systèmes de recherche d'informations [18].

Il existe bien sûr une vaste littérature sur l'utilisation des thésaurus en NLP [90].

5.3.2 AGROVOC

AGROVOC [1] est un thésaurus agricole multilingue développé par l'Organisation des Nations Unies pour l'alimentation et l'agriculture en anglais (Food and Agriculture Organization of the United Nations, FAO). Il s'agit d'un thésaurus agricole dans son sens le plus large, couvrant non seulement l'agriculture mais aussi la pêche, la foresterie, la nutrition humaine, l'environnement et la terminologie pertinente dans les sciences biologiques, physiques et sociales [79]. AGROVOC a été initié en anglais, ou du moins était orienté vers les langues d'Europe occidentale (principalement anglais, français et espagnol). Cependant, au fil des ans, le besoin de le traduire en plusieurs langues est apparu important, en particulier en chinois et en arabe qui sont des langues officielles de la FAO. Et les traductions ont ensuite été étendues à de nombreuses autres langues.

AGROVOC fournit un moyen d'organiser les connaissances permettant une gestion aisée des informations. Il s'agit d'une collection structurée de concepts, termes, définitions et relations. Les concepts représentent n'importe quel élément dans l'alimentation et l'agriculture, comme le maïs (comme le montre la figue 5.1), la faim, l'aquaculture, les chaînes de valeur ou la foresterie. Ces concepts sont utilisés pour identifier sans ambiguïté les ressources, permettant des processus d'indexation standardisés, rendant la recherche plus efficace. Chaque concept dans AGROVOC a également des termes utilisés selon plusieurs langues, les soi-disant lexicalisations. Aujourd'hui, AGROVOC se compose de plus de 37 500 concepts et plus de 750 000

1. http ://www.fao.org/agrovoc/

termes utilisent jusqu'à 40 langues. AGROVOC est le plus grand thésaurus publié sous forme de données ouvertes liées à l'alimentation et l'agriculture.

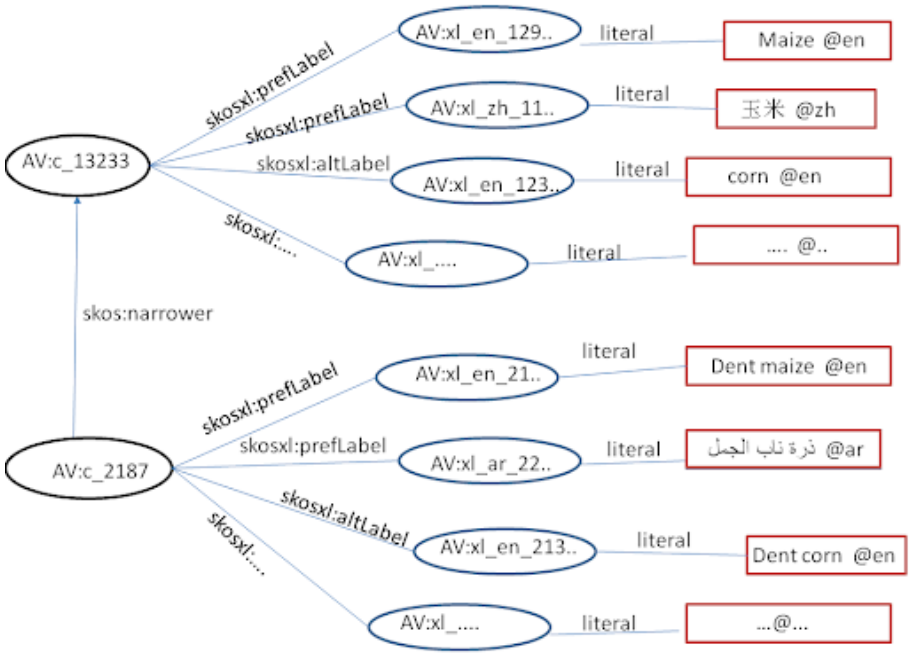

FIGURE 5.1: Exemple de recherche de mot 'Maize' en différentes langues à partir d'AGROVOC

5.3.3 Agrotagger

Utilisé pour l'indexation des ressources d'information, Agrotagger [2] est un extracteur de mots clés qui utilise le thésaurus AGROVOC comme ensemble de mots-clés admis, il peut extraire à partir de documents, de fichiers PDF et de pages web. Cet outil utilise des algorithmes de stemming et POS tagging (voir section 2.2). Agrotagger est capable d'indexer également des textes dans différentes langues.

5.3.4 Similarité Cosinus

La similarité cosinus (ou mesure cosinus) permet de calculer la similarité entre deux vecteurs à n dimensions en déterminant le cosinus de l'angle entre eux. Cette métrique est fréquemment utilisée en fouille de texte [150]. Étant donné deux vecteurs à n dimensions $\vec{v}$ et $\vec{w}$, la similarité cosinus entre eux est calculée comme suit :

$$Cosine(\vec{v}, \vec{w}) = \frac{\vec{v} \cdot \vec{w}}{|\vec{v}||\vec{w}|} = \frac{\sum_{i=1}^{n} v_i w_i}{\sqrt{\sum_{i=1}^{n} v_i^2}\sqrt{\sum_{i=1}^{n} w_i^2}} \tag{5.1}$$

La valeur qui en résulte va de 0 (signifiant des vecteurs totalement différents, c'est-à-dire utilisent du vocabulaire diffèrent), à 1 (signifiant des vecteurs identiques, c'est-à-dire utilisent exactement le même vocabulaire).

2. http ://aims.fao.org/fr/agrotagger

5.4 Description global du système

Dans cette section, nous décrivons les détails de l'approche proposée consistant à mesurer la similarité sémantique entre deux textes de qualité différente. Avant de mesurer la similarité entre les textes, une pondération des descripteurs linguistiques est classiquement mise en place. La pondération des termes permet d'identifier leur importance dans le texte.

En général, l'idée de base est d'attribuer des poids aux termes en utilisant des informations statistiques telle que la fréquence dans un texte ou relativement à un corpus dans son ensemble (voir la section 3.2 qui décrit les techniques de représentation des données). Dans notre approche, étant donné que nous traitons des textes très différents (des textes longs mais également très courts), nous avons adopté une pondération différente. Nous cherchons à donner un poids plus élevé à des termes véhiculant une certaine sémantique (illustré par leur appartenance à la ressource AGROVOC). Les thésaurus sont largement utilisés pour l'estimation de similarité comme WordNet qui modélise la connaissance lexicale en anglais [117].

Dans notre approche, la méthode suivie pour estimer la similarité sémantique entre les paires de textes passe par trois étapes comme le montre la figure 5.2. Dans un premier temps chaque document est représenté par un ensemble de termes. À partir de ces termes, une deuxième phase pondère ces termes en des vecteurs numériques. La dernière étape est l'évaluation de la similarité des textes en utilisant la mesure de similarité cosinus.

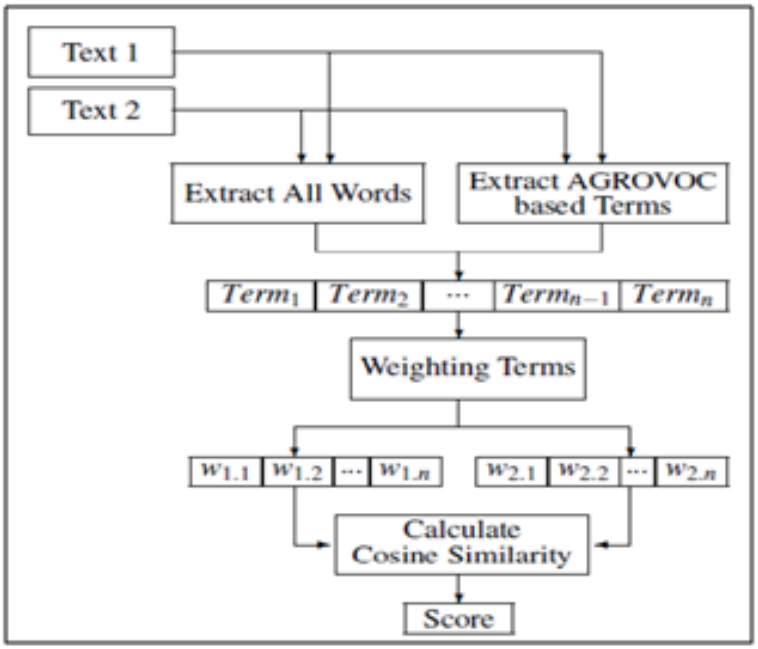

FIGURE 5.2: Application de l'approche MIGHT pour deux textes

Ces trois étapes sont détaillées dans les sous-sections suivantes.

5.4.1 Extraction

La première phase de cette approche consiste à extraire les termes du corpus selon différentes stratégies :

— La première consiste à appliquer trois techniques de pré-traitement : tokenisation, suppression des nombres et suppression des mots vides (voir section 2.2) afin d'extraire tous les mots contenus dans un texte donné.

— La deuxième étape est fondée sur l'extraction de termes avec AgroTagger. Le résultat est composé des mots clés contenus dans AGROVOC.

La première stratégie est concentrée sur les informations lexicales contenues dans les textes alors que la seconde stratégie s'intéresse aux informations sémantiques via AGROVOC.

5.4.2 Pondération

La méthode de pondération proposée consiste à construire une base représentant l'ensemble des termes des corpus. Ensuite, deux vecteurs de poids (pour chaque pair de texte) sont construits de cette manière : tous les mots identifiés dans un texte par rapport à une base représentant l'ensemble des descripteurs des corpus sont pondérés à 1, les termes qui sont extraits avec l'extracteur d'AGROVOC sont pondérés à 2 et les descripteurs linguistiques identifiés sur la base de ces deux méthodes sont pondérés à 3.

5.4.3 Évaluation

Enfin, une mesure de similarité (cosinus) calcule la proximité entre deux textes donnés à partir des vecteurs pondérés. Afin d'évaluer l'approche proposée, nous avons développé un logiciel dédié.

5.5 Expérimentations

Dans cette section, nous montrons comment appliquer notre approche sur des ensembles de données hétérogènes (tweets et articles scientifiques) en langue française relativement à la thématique «changement climatique».

5.5.1 Corpus et protocole expérimental

Dans ces expérimentations, nous avons collecté 4 corpus différents :

— Dans un premier temps, nous avons recueilli des tweets en suivant les comptes Twitter avec des hashtags spécifiques : #réchauffementClimatique, #changementClimatique. Ainsi, nous avons constitué un corpus de tweets français issus d'associations, d'organisations, de célébrités et de citoyens abordant cette thématique.

— Puis un autre corpus de tweets non liés à ce domaine a été utilisé, Politweets [110], rassemble les tweets de 7 personnalités de 6 différents groupes politiques français. Des twittos en français envoyés en 2013-14 sont extraits des comptes Twitter de ces personnalités , ce qui fait le corpus total de 34273 messages (tweets) qui contiennent 502 085 *tokens*, la ponctuation est exclue.

— Des articles scientifiques traitant du changement climatique ont été mobilisés. Ces données sont des résumés en français d'articles, de livres, de chapitres de livres, de thèses, etc., à partir d'Agritrop [3], archive ouverte du Cirad [4].

— Des articles scientifiques non liées au changement climatique, obtenus à partir d'une collection de résumés en français d'articles de TETIS [5] (Territoires, environnement, télédétection et information spatiale).

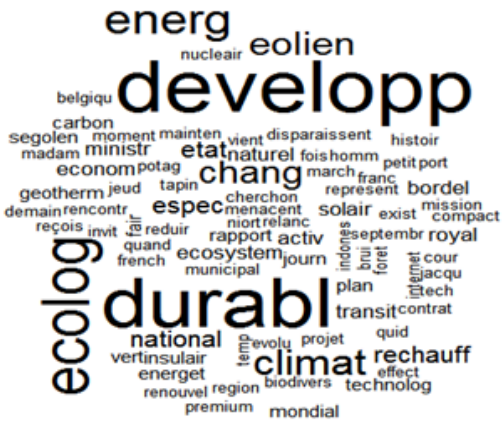

FIGURE 5.3: Nuage de mots du corpus de tweets liés au thème "changement climatique"

FIGURE 5.4: Nuage de mots du corpus de tweets non liés au thème "changement climatique"

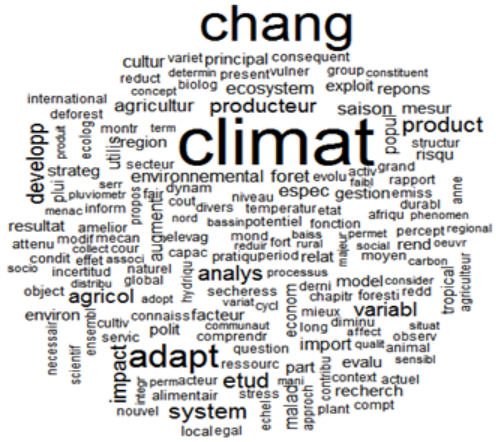

FIGURE 5.5: Nuage de mots du corpus d'articles scientifiques liés au thème "changement climatique"

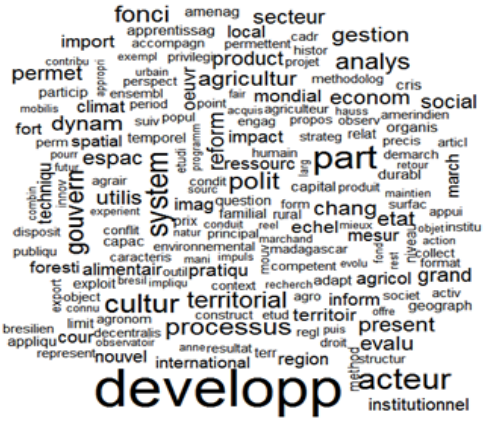

FIGURE 5.6: Nuage de mots du corpus d'articles scientifiques non liés au thème "changement climatique"

Ces 4 collections de données textuelles sont notées de la manière suivante : CT : Tweets traitant du changement climatique, NCT : Tweets non liés au changement climatique (Politweets), CA : Articles scientifiques traitant du changement climatique (Cirad), NCA : Articles scientifiques non liés au changement climatique (TETIS).

3. http ://agritrop.cirad.fr/
4. http ://www.cirad.fr/
5. https ://tetis.teledetection.fr/index.php/fr/tetissummary

5.5.2 Résultats

Les expérimentations sont réalisées selon 10 itérations, pour chacune d'elle, nous avons sélectionné aléatoirement N paires de textes (N = 10). Pour chaque paire l'approche MIGHT a été appliquée. Le nombre d'exécutions total est de 3000. Le résultat final représente la moyenne des résultats cosinus obtenus pour chaque itération. Le tableau 5.1 montre les résultats obtenus afin de comparer CT et CA, CT et NCT, CT et NCA.

Itération	**CT/NCT**	**CT/CA**	**CT/NCA**
1	0.0073	0.0025	0.0078
2	0	0.0063	0
3	0	0.0128	0.0093
4	0	0	0
5	0	0	0
6	0	0.0182	0.032
7	0	0.0069	0
8	0	0.0182	0.032
9	0	0.0069	0
10	0	0.0106	0

TABLE 5.1: Résultats expérimentaux de l'approche MIGHT

Les degrés de similarité les plus élevés pour cinq itérations est la similarité entre CT et CA (couvrant des sujets proches) mettent en avant des premiers résultats encourageants restitués par notre approche MIGHT.

5.6 Conclusion

Dans ce chapitre, nous avons abordé la question de l'évaluation de la similarité sémantique entre des documents de nature différente mais qui peuvent porter sur des sujets proches. Notre approche est fondée sur l'extraction de descripteurs linguistiques issus d'un texte (mots) et des termes (mots et syntagmes) propres à un thésaurus en appliquant une pondération sémantique spécifique. Notre méthode a tendance à rapprocher des textes ayant des thématiques proches ce qui permet de mettre en relation des données de qualité différente. De nombreuses perspectives peuvent être proposées comme l'élimination du vocabulaire spécifique aux tweets (phrases ou expressions spécifiques), l'expansion de contextes (par exemple, en considérant l'ensemble de tweets écrits par le même auteur dans une même fenêtre temporelle).

Le chapitre suivant décrit une nouvelle contribution qui consiste à sélectionner les meilleurs descripteurs des documents textuels en se basant sur des algorithmes d'optimisation bio-inspirés.

Chapitre 6

Une approche bio-inspirée de sélection des caractéristiques pour faire correspondre des documents texte

Sommaire

6.1 Introduction

Afin d'évaluer la similarité entre les documents, on utilise généralement une représentation de documents très connue qui est le modèle du BoW. Comme décrit dans la section 3.2, ce modèle crée une matrice avec le nombre de mots pour chaque instance de données (c'est-à-dire les documents). Dans ce modèle, un texte est représenté comme le sac de ses mots, sans tenir compte de la grammaire et même de l'ordre des mots. Si l'on utilise directement le vocabulaire contenu dans les textes d'apprentissage, on se retrouve avec un espace vectoriel de très grande dimension. Chaque texte sera représenté par un vecteur avec autant de termes qu'il y a dans le vocabulaire.

Le traitement de l'espace vectoriel demanderait beaucoup de mémoire et de temps de calcul et pourrait nous empêcher d'utiliser des algorithmes de traitement plus complexes [137]. La sélection des caractéristiques est une technique de prétraitement couramment utilisée pour les données de grande dimension (voir section 4.2). Cela implique la sélection d'un sous-ensemble de caractéristiques pertinentes et la suppression des caractéristiques non pertinentes, redondantes et bruitées, pour une représentation des données plus facile et plus précise.

Ce chapitre de thèse est organisé comme suit. Les motivations de cette contribution sont présentées en section 6.2. L'état de l'art des travaux réalisés pour l'utilisation des algorithmes bio-inspirés comme outil de sélection de caractéristiques dans l'exploration de données sont présentés en section 6.3. En section 6.4, nous présentons les éléments de contributions utilisés dans nos travaux. L'approche proposée est présentée en section 6.5. Une description des expérimentations et des résultats obtenus est présentée en section 6.6. Enfin, nous terminerons ce chapitre par une conclusion et quelques perspectives qui sont présentées en section 6.7.

6.2 Motivation

En raison de l'énorme augmentation de la quantité de données, le problème de sélection des caractéristiques est connu pour être un problème combinatoire, et la mise en oeuvre d'une approche d'optimisation combinatoire permet de rechercher efficacement l'ensemble minimum de caractéristiques idéalement nécessaires et suffisants pour décrire la sémantique d'un ensemble de documents texte ; et cela afin de réduire le coût et d'augmenter l'exactitude de correspondance de ces documents.

Dans les applications du monde réel, les utilisateurs sont en général plus intéressés à obtenir de bonnes solutions dans un laps de temps raisonnable plutôt que de privilégier l'obtention des solutions optimales. Par conséquent, nous privilégions les méthodes métaheuristiques qui se sont avérées efficaces pour traiter des applications du monde réel [167]. Elles nous permettent d'obtenir des solutions raisonnablement bonnes, sans avoir besoin d'explorer tout l'espace de solutions.

Parmi les métaheuritiques basées population, GA (voir section 4.4.2.1) qui est l'un des algorithmes populaires d'optimisation, c'est une stratégie de recherche faisant partie des algorithmes évolutionnaires à base de population, développé dans les années 70 par John Holland et son équipe [71]. Il est basé sur la reproduction et l'évolution naturelle des individus en s'inspirant du principe de l'évolution darwinienne des populations biologiques. GA a montré avec succès sa grande capacité à résoudre des problèmes de sélection des caractéristiques optimales [166, 81, 128, 17].

Cependant, il a montré certaines limites telles que le stockage important des calculs, la non-convergence vers un optimum global et la convergence prématurée [39].

Pour dépasser ces limites, l'émergence d'une tendance visant à intégrer les concepts quantiques aux algorithmes conventionnels, a conduit à une nouvelle version de GA. Le calcul quantique est une science interdisciplinaire issue de la science de l'information et de la science quantique. Le premier algorithme quantique a été proposé par Shor [147], pour la factorisation des nombres. Grover [64] a également proposé un algorithme quantique pour la recherche aléatoire dans les bases de données, la complexité de son algorithme a été réduite pour être de l'ordre de $O(\sqrt{N})$ alors que celle des algorithmes classiques est $O(N)$. Depuis la fin des années 90, la fusion de l'informatique quantique et du calcul évolutif s'est avérée prometteuse pour résoudre des problèmes complexes. Le premier travail d'hybridation de ces deux paradigmes remonte à 1996 [125], lorsque Narayanan et Moore ont proposé le premier algorithme évolutionnaire inspiré du quantique Quantum Inspired Evolutionary Algorithm (QIEA).

Dans notre étude nous proposons d'adapter l'algorithme génétique inspiré du quantique pour une recherche efficace des sous-ensembles optimales des caractéristiques des documents textuels. Cet algorithme a prouvé son efficacité dans de nombreuses études [15, 136, 5] permettant de tirer partie de GA :

— Traiter plusieurs données en parallèle en utilisant le minimum d'informations,

— Assurer un bon équilibre entre l'exploration et l'exploitation de l'espace de recherche.

et de l'informatique quantique :

— Une représentation quantique des individus ; ce qui permet une exploration plus poussée,

— Une bonne couverture, elle peut couvrir une grande partie de l'espace de recherche,

— Une complexité réduite ; très peu d'individus sont nécessaires pour une bonne représentation de l'espace de recherche,

— Une énorme puissance de calcul grâce à la superposition d'états et aux opérateurs quantiques, permettant le traitement d'une grande quantité d'informations en parallèle.

6.3 Travaux connexes

Dans cette section, nous allons présenter les travaux connexes liés à l'utilisation des algorithmes bio-inspirés comme outil de sélection de caractéristiques dans l'exploration de données.

Pour des raisons d'efficacité, les algorithmes bio-inspirés ont été largement utilisés comme outil de sélection de caractéristiques dans l'exploration de données. Dans les travaux présentés dans [82], les auteurs ont proposé un algorithme d'optimisation hybride des colonies de fourmis ACO pour la sélection de caractéristiques, appelé Ant Colony Optimization for Feature Selection (ACOFS), en utilisant un réseau neuronal. Un aspect clé de cet algorithme est la sélection d'un sous-ensemble de caractéristiques saillantes de taille réduite. ACOFS utilise une technique de recherche

hybride qui combine les avantages des approches d'emballage et de filtrage. Afin de faciliter la recherche hybride, les auteurs ont conçu de nouveaux ensembles de règles pour la mise à jour des phéromones et une mesure heuristique de l'information. D'autre part, les fourmis sont guidées dans des directions correctes, tout en construisant des chemins de graphique (sous-ensemble) en utilisant un schéma borné à chaque étape de l'algorithme. Les combinaisons ci-dessus fournissent finalement non seulement un équilibre efficace entre l'exploration et l'exploitation des fourmis dans la recherche, mais intensifient également la capacité de recherche globale de l'ACO pour une solution de haute qualité dans la sélection des caractéristiques. Il existe d'autres études qui ont appliqué les algorithmes des colonies de fourmis au problème de la sélection des caractéristiques telles que [14, 3].

Zahran et Kanaan [168] ont introduit un algorithme de sélection de caractéristiques basé sur Particle Swarm Optimization (PSO) pour améliorer les performances de la catégorisation de texte arabe. Ils ont utilisé des réseaux Radial Basis Function (RBF) comme classifieur de texte. Sur la base du même algorithme bio-inspiré, Xue[164] a proposé deux algorithmes multi-objectifs pour sélectionner le front de Pareto des solutions non dominées (sous-ensembles de caractéristiques) pour la classification. Le premier algorithme introduit l'idée d'un GA multi-objectif basé sur un tri non dominé dans PSO pour la sélection de caractéristiques. Dans le deuxième algorithme, le PSO multi-objectif utilise les idées d'encombrement, de mutation et de dominance pour rechercher les solutions du front de Pareto.

Siedlecki et Sklansky dans [148] ont introduit l'utilisation de l'algorithme génétique pour la sélection des caractéristiques. Dans une approche GA, un sous-ensemble de caractéristiques donné est représenté comme une chaîne binaire 'Chromosome' de longueur n, avec un zéro ou un en position i indiquant l'absence ou la présence de la caractéristique i dans l'ensemble, respectivement. Il faut noter que n est le nombre total de caractéristiques disponibles. Une population de chromosomes est maintenue. Chaque chromosome est évalué pour déterminer son $fitness$, qui détermine la probabilité que le chromosome survive et se reproduise dans la prochaine génération. De nouveaux chromosomes sont créés à partir d'anciens chromosomes par les processus suivants : (1) croisement, où des parties de deux chromosomes parents différents sont mélangés pour créer une progéniture et (2) mutation, où les bits d'un seul parent sont perturbés au hasard pour créer un enfant [167]. Jourdan et al. [81] ont également présenté un GA dédié à un problème de sélection de caractéristiques, mais dans un cas particulier rencontré dans l'analyse génétique de différentes maladies. La spécificité de ce problème est que les auteurs ne recherchent pas une seule caractéristique, mais plusieurs associations de caractéristiques pouvant être impliquées dans la maladie étudiée. Il existe d'autres études appliquant GA sur le problème de la sélection des caractéristiques [166, 128, 17].

6.4 Éléments de contribution

Dans cette section, nous allons présenter les éléments utilisés dans notre contribution concernant la sélection de caractéristiques.

6.4.1 Algorithme Génétique Inspiré du quantique

Quantum Inspired Genetic Algorithm (QIGA) est un GA enrichi par les concepts et principes de l'informatique quantique, tels que le qubit, la superposition d'états, la mesure quantique et les opérateurs quantiques. En informatique quantique, la plus petite unité de stockage d'informations est le qubit. Un qubit peut être à l'état '1', à l'état '0' ou dans une superposition des deux. L'état d'un qubit peut être représenté comme indiqué par la formule 6.1.

$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{6.1}$$

Où $|0\rangle$ et $|1\rangle$ représentent les valeurs conventionnelles des bits 0 et 1, respectivement. α et β sont des nombres complexes satisfaisant :

$$|\alpha|^2 + |\beta|^2 = 1 \tag{6.2}$$

où $|\alpha|^2$ représente la probabilité que le qubit soit trouvé dans l'état '0' et $|\beta|^2$ représente la probabilité que le qubit soit trouvé dans l'état '1'. L'élément de base étant ici le qubit $\binom{\alpha}{\beta}$, un chromosome est simplement une chaîne de m qubits formant un registre quantique comme le monter la Figure 6.1. L'ensemble des chromosomes quantiques représente une population, cette dernière peut être initialisée d'une manière aléatoire, ou selon Han et Kim [66], la manière la plus simple de créer la population initiale est d'initialiser toutes les amplitudes de qubits par la valeur $\frac{1}{\sqrt{2}}$. Cela signifie qu'un chromosome représente tous les états de superposition quantique avec une probabilité égale.

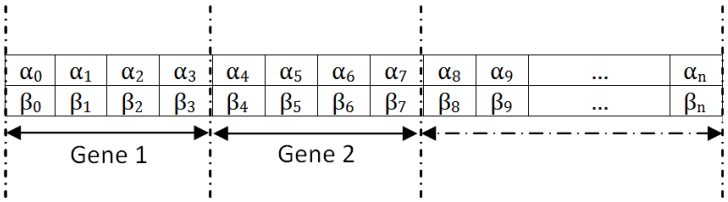

FIGURE 6.1: Structure d'un chromosome quantique

En informatique classique, les états possibles d'un système de n bits forment un espace vectoriel de n dimensions, c'est-à-dire que nous avons 2^n états possibles. Cependant, dans un système quantique de n qubits, l'espace d'états résultant a des dimensions 2^n. C'est cette croissance exponentielle de l'espace d'états avec le nombre de particules qui suggère une possible accélération exponentielle du calcul sur les ordinateurs quantiques par rapport aux ordinateurs classiques. La base de l'espace d'états d'un système quantique de n qubits est : $\{|00...0\rangle, |00...1\rangle \ldots |11...1\rangle\}$ [155, 49].

Mesure quantique : Pour extraire un chromosome classique d'un chromosome quantique, nous appliquons la mesure quantique qui projette l'état quantique sur l'un des états de base associés à l'appareil de mesure. Le résultat dépend des amplitudes du qubit, comme un qubit dont la valeur $|\alpha|^2 = 0.8$ aura 80 % de chances d'être un '1' et 20 % d'être dans l'état '0'. La mesure à plusieurs qubits peut être traitée comme une série de mesures à un qubit dans la base standard. La Figure 6.2 donne une illustration de mesure quantique appliquée à un chromosome quantique.

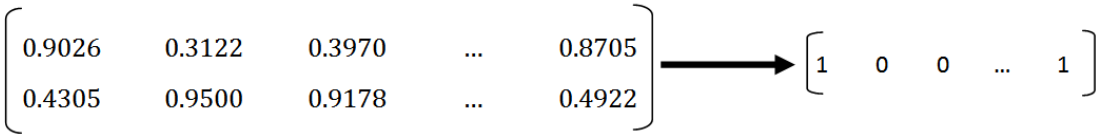

FIGURE 6.2: Mesure quantique

Croisement quantique : Le croisement quantique a le même principe qu'un croisement conventionnel. Mais il fonctionne sur les chromosomes quantiques. C'est donc un croisement de matrice de probabilité qui génère en conséquence de nouvelles matrices de probabilité. Le croisement quantique entre deux individus (parents) à un moment donné peut générer deux nouveaux individus (progéniture) dont les gènes proviennent des deux parents.

Mutation quantique : La mutation classique fonctionne comme une petite perturbation qui inverse le bit muté. Dans une mutation quantique, il y a aussi une perturbation, mais elle fonctionne sur les probabilités d'un qubit du chromosome concerné, comme suit. Considérons un qubit $|A\rangle = \alpha|0\rangle + \beta|1\rangle$. La mutation quantique qubit de A génère le qubit $|B\rangle = \beta|0\rangle + \alpha|1\rangle$, comme il est indiqué dans la Figure 6.3.

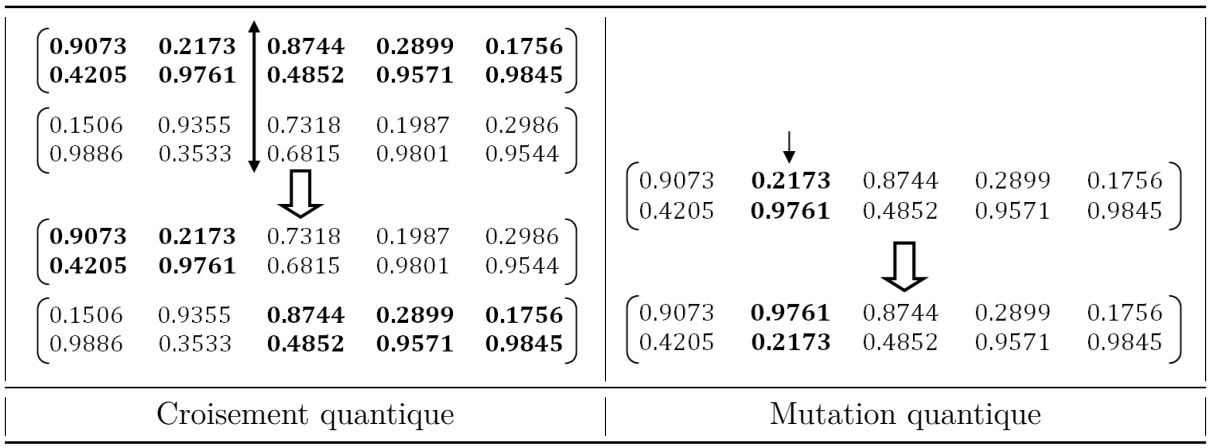

FIGURE 6.3: Opérateurs génétiques de QIGA.

Interférence quantique : À chaque itération, la meilleure solution actuelle sert de guide pour trouver de nouvelles solutions qui pourraient être meilleures (voir Figure 6.4). Ceci est implémenté via une porte quantique appelée la porte D qui prend la forme matricielle donnée dans la formule 6.3 :

$$D = \begin{pmatrix} \cos\delta\theta & -\sin\delta\theta \\ \sin\delta\theta & \cos\delta\theta \end{pmatrix} \tag{6.3}$$

La variable $\delta\theta$ représente un angle de rotation qui fait converger le qubit vers l'état 0 ou 1 selon son signe objectif, la porte D est appliquée sur un qubit $\binom{\alpha}{\beta}$, le

qubit $\begin{pmatrix}\alpha' \\ \beta'\end{pmatrix}$ mis à jour est obtenu en appliquant la formule 6.4 :

$$\begin{pmatrix}\alpha' \\ \beta'\end{pmatrix} = D \cdot \begin{pmatrix}\alpha \\ \beta\end{pmatrix} \tag{6.4}$$

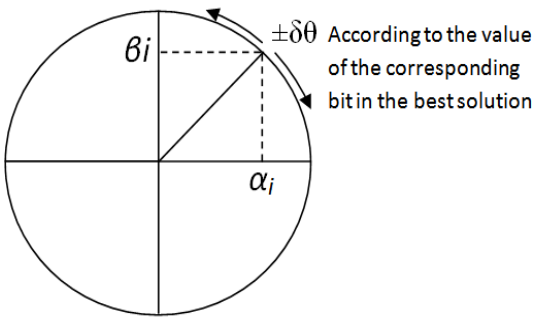

FIGURE 6.4: Interference quantique

α	β	Valeur de bit de référence	Angle
> 0	> 0	1	$+\delta\theta$
> 0	> 0	0	$-\delta\theta$
> 0	< 0	1	$-\delta\theta$
> 0	< 0	0	$+\delta\theta$
< 0	> 0	1	$-\delta\theta$
< 0	> 0	0	$+\delta\theta$
< 0	< 0	1	$+\delta\theta$
< 0	< 0	0	$-\delta\theta$

TABLE 6.1: Table de consultation de l'angle de rotation.

Sélection : Généralement, toutes les méthodes de sélection de GA conventionnel sont applicables sur QIGA. Comme la sélection par rang, par roulette ou par tournoi. Elles visent toutes à sélectionner les meilleurs individus pour le prochain processus de reproduction en se basant sur différentes stratégies tels que le classement et le tournoi.

6.4.2 Rang Réciproque Moyen

La mesure d'extraction d'informations de rang réciproque ou ce que l'on appelle en anglais Reciprocal Rank (RR) calcule l'inverse du rang auquel le premier document pertinent a été récupéré. RR vaut 1 si un document pertinent a été récupéré au rang 1, sinon il vaut 0,5 si un document pertinent a été récupéré au rang 2 et ainsi de suite. Lorsqu'elle est calculée en moyenne sur toutes les requêtes, la mesure est appelée le rang réciproque moyen ou ce que l'on appelle en anglais Mean Reciprocal Rank (MRR) [38]. Cette mesure statistique (voir Formule 6.5) évalue tout processus qui produit une liste de réponses possibles à un échantillon de requêtes, triées par la probabilité d'exactitude.

$$MRR = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i} \tag{6.5}$$

Où $rank_i$ fait référence à la position de classement du premier document pertinent pour la $i^{ème}$ requête Q.

6.4.3 Validation croisée

La validation croisée est une méthode statistique d'évaluation et de comparaison des algorithmes d'apprentissage en divisant les données en deux segments : l'un utilisé pour apprendre ou entraîner un modèle et l'autre utilisé pour valider le modèle. Dans une validation croisée typique, les ensembles d'apprentissage et de validation doivent se croiser en cycles successifs de sorte que chaque point de données ait une

chance d'être validé. La forme de base de la validation croisée est la validation croisée k fois. Dans la validation croisée de k fois, les données sont d'abord partitionnées en k segments ou plis de taille égale (ou presque égale). Par la suite, k itérations d'apprentissage et de validation sont effectuées de telle sorte que dans chaque itération un pli différent des données est mis à l'écart pour la validation tandis que les $k-1$ plis restants sont utilisés pour l'apprentissage [140]. La figure 6.5 montre un exemple avec $k = 3$. Les sections claires des données sont utilisées pour l'apprentissage tandis que les sections les plus sombres sont utilisées pour la validation.

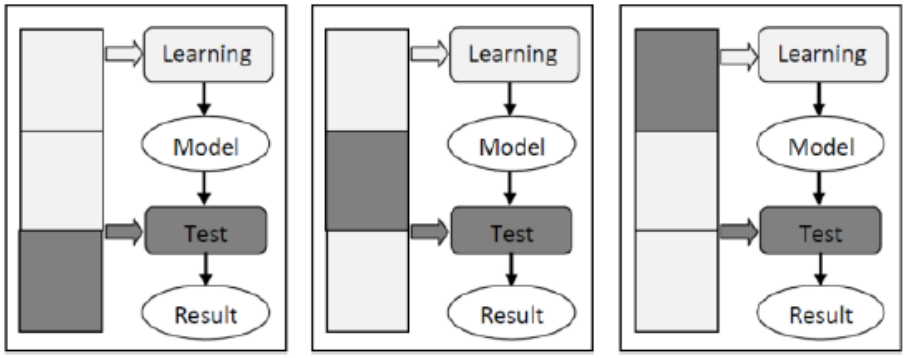

FIGURE 6.5: Procédure de validation croisée / k=3.

6.5 Description globale du système

Comme nous l'avons indiqué en section 1.4, les approches d'espaces vectoriels qui servent à l'évaluation de la similarité textuelle, sont généralement composée de trois modules : pré-traitement, sélection de caractéristiques et mise en correspondance. La figure 6.6 montre l'architecture de l'approche proposée.

6.5.1 Pré-traitement des données

Le pré-traitement de texte (étape 1, Figure 6.6) est une tâche qui joue un rôle très important dans les techniques et les applications de fouille de texte. Notre approche bio-inspirée étant destinée à traiter des documents textuels hétérogènes, le jeu de données collecté utilisé dans nos travaux est constitué d'articles scientifiques, articles de blog et tweets. Nous avons adopté les techniques de pré-traitement les plus utilisées dans la littérature [52, 158, 130] qui sont le nettoyage et le stemming.

Afin de nettoyer le corpus d'articles scientifiques et de articles de blog, les tâches effectuées sont les suivantes :

— Suppression des liens, de la ponctuation, des chiffres et des espaces inutiles,

— Suppression des mots vides à l'aide d'une liste standard que nous avons enrichie,

— Désaccentuation (concernant les mots en français),

— Stemming,

— Concernant les tweets, nous avons supprimé des hashtags et les citations.

Figure 6.7 montre un exemple de *tokens* obtenus après l'étape de pré-traitement.

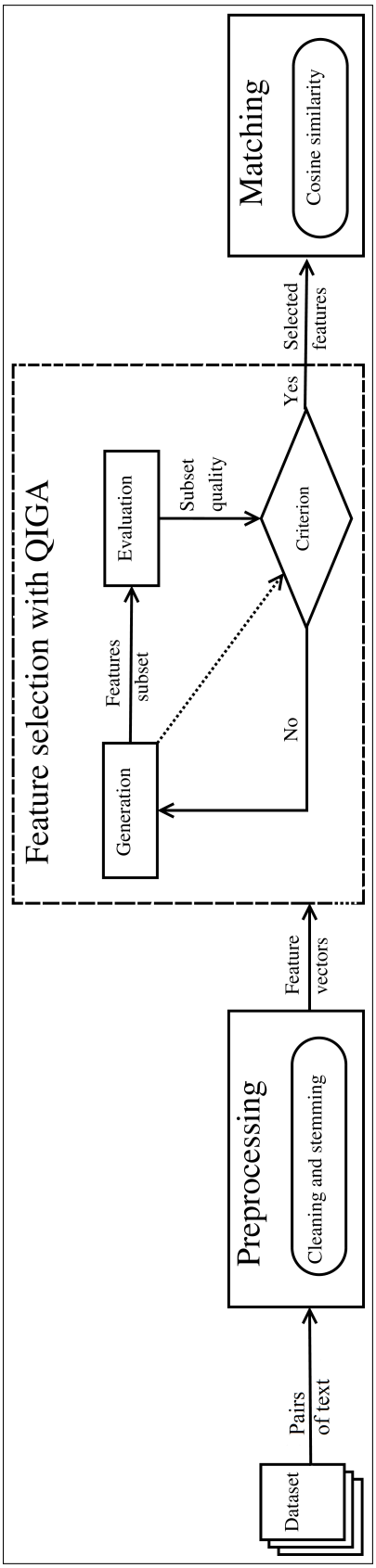

FIGURE 6.6: L'approche bio-inspirée non supervisée proposée pour faire correspondre les documents textuels

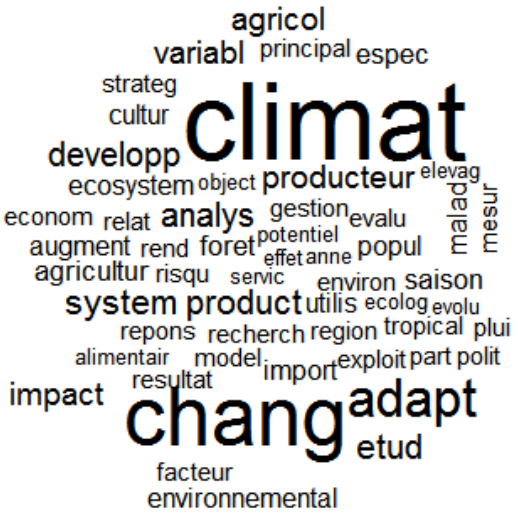

FIGURE 6.7: Nuage de mot d'un corpus prétraité.

6.5.2 Sélection des caractéristiques

Dans cette contribution, nous avons utilisé QIGA (décrit précédemment dans la section 6.4) dans le but de sélectionner le minimum des caractéristiques tout en maximisant le degré d'exactitude d'évaluation de la similarité sémantique entre les différents documents textuels (étape 2, Figure 6.6). La présentation des solutions ainsi que la fonction objectif sont modélisées comme suit.

6.5.2.1 Présentation de la solution

Dans notre approche bio-inspirée, une représentation quantique des solutions est adoptée. Un sous-ensemble de caractéristiques donné est représenté par un registre quantique «chromosome quantique» de longueur n (voir la figure 6.1), où chaque qubit est composé de deux valeurs, à savoir α et β en position i indiquant l'absence ou la présence de la caractéristique i dans l'ensemble. Il faut noter que n est le nombre total de caractéristiques disponibles. Une population est simplement un ensemble de chromosomes quantiques.

6.5.2.2 Fonction objectif

En général, la fonction objectif est constituée de deux termes qui sont en concurrence l'un avec l'autre : le nombre de caractéristiques (à minimiser) et la qualité (à maximiser), la décision est un compromis entre ces deux objectifs [14, 3, 82]. Dans notre approche, la qualité est évaluée avec la mesure du Cosinus (décrite dans la section 5.3.4) entre le vecteur de caractéristiques à réduire et les vecteurs d'apprentissage. Afin de calculer la similarité Cosinus, nous devons passer par une étape de pondération, dans laquelle nous avons utilisé deux méthodes :

— La première méthode proposée consiste à concaténer les deux vecteurs de caractéristiques (c'est-à-dire le vecteur à optimiser et le vecteur utilisé pour l'apprentissage). Ensuite, deux vecteurs de poids P1 et P2 sont construits de cette manière : pour chaque caractéristique. Si elle n'existe pas dans le vecteur à

optimiser (pour P1, mais pour P2, l'existence du terme est vérifiée dans le vecteur d'apprentissage) le poids donné est '0'. Sinon, et s'il existe dans le vecteur pour optimiser uniquement, le poids donné est' 1 ' ; s'il existe dans les deux vecteurs, le poids donné est' 2 '.

— La deuxième méthode de pondération est basée sur la mesure TF-IDF décrite dans la section 3.2.3.

Ensuite, la fitness est calculée à l'aide de la formule 6.6.

$$Fitness = \alpha \cdot Taille + (1 - \alpha) \cdot Qualité \tag{6.6}$$

La démarche de calcul de la fonction objectif est décrite dans l'algorithme 2, et les principales étapes de la phase de sélection des caractéristiques sont décrites dans l'algorithme 3 :

Algorithm 2 : Fonction Objectif

Entrée : Chromosome à réduire + Chromosome d'apprentissage
Sortie : Fitness de chaque chromosome

Début
  Pour chaque chromosome quantique **Faire**
    Appliquer la mesure quantique
    Calculer le nombre d'entités non sélectionnées : *zéros*
    Concaténer les vecteurs de caractéristiques (sélectionné + apprentissage)
    Générer deux vecteurs de poids : $P1$ et $P2$
    Calculer la similarité Cosinus entre $P1$ et $P2$: cos
    Calculer : $Fitness = \alpha \cdot zéros + (1 - \alpha) \cdot \cos$
  Fin Pour
Fin

Algorithm 3 : QIGA pour la sélection des caractéristiques

Entrée : vecteurs des caractéristiques
Sortie : vecteurs des caractéristiques sélectionnées

Début
  Initialisation
  Génération des solution et évaluation des chromosomes quantiques
  Sélection de la meilleur solution
  Tant que Nombre d'itérations <= Max **Faire**
    Application des opérateurs génétiques quantiques
    Génération des solution et évaluation des chromosomes quantiques
    Sélection de la meilleure solution
    Application de l'interférence quantique
  Fin Tant que
Fin

Dans ce qui suit, une description plus détaillée sur le déroulement de QIGA est donnée.

— Initialisation : dans la première étape, l'algorithme détermine la taille de la population de chromosomes quantiques, définit les valeurs α et β de la population initiale de manière aléatoire, définit la probabilité de mutation quantique et de croisement et détermine le nombre maximum des itérations.
— Génération de solution et évaluation des chromosomes quantiques : Dans cette étape, l'opération de mesure est appliquée sur chaque chromosome pour en avoir une solution parmi toutes celles présentées en superposition, puis la qualité des solutions obtenues est évaluée en appliquant la fonction objectif.
— Sélectionner le meilleur sous-ensemble : Cette étape sert à trier les sous-ensembles sélectionnés en fonction de leurs valeurs de fitness, sélectionner les meilleurs sous-ensembles et enregistrer le meilleur chromosome quantique.
— Effectuer des opérateurs génétiques quantiques : Pour assurer une perturbation à tous les chromosomes, un opérateur de croisement quantique est appliqué sur la population quantique actuelle, puis un opérateur de mutation quantique est appliqué sur la population croisée.
— Appliquer l'interférence quantique : pour améliorer la qualité des solutions, chaque qubit est décalé dans le sens de la valeur de bit correspondant dans la meilleure solution actuelle.

6.5.3 Mise en correspondance

Afin de faire correspondre les documents textuels hétérogènes (étape 3, Figue6.6), nous calculons la similarité Cosinus en utilisant deux méthodes. La première consiste à calculer le Cosinus entre les vecteurs BoW, et la seconde est un calcul Cosinus entre vecteurs TF-IDF de toutes les paires de tests sur la base des caractéristiques sélectionnées lors de la phase précédente.

6.6 Étude expérimentale

Dans cette section, l'approche proposée est validée en l'appliquant sur trois types de documents texte : articles scientifiques, articles de blog et tweets. Ensuite, la mesure MRR est adoptée afin de comparer l'approche proposée à l'approche classique du BoW.

6.6.1 Description du jeu de données

Dans ce travail, nous nous concentrons sur l'étude de la variété des données textuelles liées au thème du changement climatique. Par conséquent, nous avons collecté des documents hétérogènes en français de diverses sources.

— Les articles scientifiques traitent le sujet du changement climatique à partir d'une collection de résumés d'articles, de livres, de chapitres d'ouvrages, de thèses, etc., issus de l'archive ouvert Agritrop [1] du CIRAD [2] (Recherche agronomique et coopération internationale, organisation française œuvrant pour le développement durable des régions tropicales et méditerranéennes 'publications).

1. http ://agritrop.cirad.fr/
2. http ://www.cirad.fr/

— Les articles scientifiques non liés au sujet du changement climatique d'une collection de résumés du laboratoire TETIS [3] (pour les territoires, l'environnement, la télédétection et l'information spatiale). Les domaines de recherche de ces articles sont l'agroforesterie, l'urbanisation, la gestion des ressources naturelles, l'aménagement du territoire, etc.

— Des articles de blog traitant de la question climatique, collectés manuellement à partir de deux blogs [4] traitant du thème du changement climatique.

— Des tweets traitant du sujet du changement climatique de DEFITweets [5] qui est un corpus traitant du changement climatique (15 000 tweets).

— Des tweets non liés au changement climatique du corpus Politweets [110]. Ce corpus rassemble les tweets de 7 personnalités de 6 groupes politiques français différents. Extrait des comptes Twitter de ces personnes par une méthode qui sélectionne les messages envoyés en 2013 et 2014, soit un corpus total de 34273 messages (tweets).

Dans nos expériences, nous avons sélectionné aléatoirement 100 documents pour chaque catégorie. Ces données sont nommées : CA pour les articles scientifiques dans le domaine du changement climatique, NCA représente les articles scientifiques non liés au changement climatique, CB pour les articles de blog climatiques, NCB pour les articles de blog non climatiques, CT représente les tweets sur le changement climatique et NCT est pour les tweets non climatiques. Les paires pertinentes sont des paires de documents hétérogènes couvrant le même sujet, par exemple, un document de CA et un document de CB.

6.6.2 Spécification de paramètres

Après le pré-traitement des données, notre objectif est de sélectionner les caractéristiques minimales qui décrivent la sémantique du corpus en utilisant QIGA avec les paramètres suivants : le nombre d'itérations a été fixé à 300, la taille de la population a été choisie à 5, l'angle de rotation était de $\pi/40$, la probabilité de mutation et de croisement a été choisie à respectivement 0,1 et 0,9, la valeur α dans la fonction objectif a pris des valeurs de 0,1 à 0,9. Les valeurs des paramètres sont obtenues après avoir effectué plusieurs expériences sur chaque variable afin de trouver la meilleure valeur.

Concernant le dataset, comme nous utilisons une triple validation croisée (voir section 6.4.3), les données sont réparties comme suit : pour chaque catégorie, 66 documents d'apprentissage et 33 documents de test.

QIGA se déroule comme suit. Pour extraire les meilleures caractéristiques des articles climatiques CA, les entrées de l'algorithme sont le vecteur de caractéristiques CA comme vecteur à réduire et le vecteur de caractéristiques (CB + CT) comme vecteur d'apprentissage des caractéristiques.

— Tout d'abord, nous commençons par créer aléatoirement la population initiale.

— Ensuite, pour chaque itération, on applique les opérateurs génétiques.

— On évalue la population à l'aide de la formule de la fonction objectif.

3. https ://tetis.teledetection.fr/index.php/fr/tetis-summary

4. http ://www.ecoco2.com/blog/rechauffementclimatique
http ://maplanete.blogs.sudouest.fr/tag/rechauffement+climatique

5. https ://deft.limsi.fr/2015/index.php

— Et on extrait les mots choisis par l'algorithme.

Ensuite, nous appliquons les étapes ci-dessus pour extraire les meilleures caractéristiques des blogs climatiques (les entrées sont CB pour réduire et (CA + CT) comme vecteur d'apprentissage), et les meilleures caractéristiques des tweets climatiques (les entrées sont CT pour réduire et (CA + CB) comme vecteur d'apprentissage).

Enfin, la base des caractéristiques qui décrit la sémantique du corpus est la concaténation des meilleures caractéristiques sélectionnées en CA, CB et CT.

6.6.3 Résultats expérimentaux et discussion

Nous présentons dans cette section les résultats obtenus en appliquant l'approche bio-inspirée proposée sur trois catégories de documents textuels hétérogènes collectés à partir de différentes sources. Ces résultats sont comparés à ceux donnés par l'approche classique BoW. Le tableau 6.2 montre les résultats de trois variantes de l'approche proposée, qui se différencient en deux étapes : l'étape de pondération qui précède le calcul du Cosinus dans la fonction objectif et l'étape de calcul du Cosinus qui précède le calcul MRR.

— La première méthode utilise une méthode de pondération proposée, et MRR est calculé à l'aide des vecteurs d'occurrences des termes de toutes les paires de tests.

— La deuxième méthode utilise TF-IDF comme mesure de pondération, et MRR est calculé en utilisant les vecteurs d'occurrences des termes de toutes les paires de test.

— La dernière méthode utilise le TF-IDF comme mesure de pondération, et MRR est calculé en utilisant les vecteurs TF-IDF de toutes les paires de test.

Les valeurs MRR sont calculées pour :

— Les bases de caractéristiques générées selon différentes valeurs α de la fonction objectif (voir formule 6.6).

— Les caractéristiques du corpus d'apprentissage, après la phase de pré-traitement.

— Les caractéristiques d'apprentissage du corpus sans pré-traitement, seule la suppression des ponctuations est appliquée.

D'après les résultats rapportés, nous pouvons constater que la première méthode n'a pas donné de meilleurs résultats par rapport à l'approche prétraitée du BoW, mais elle a surpassé l'approche non prétraitée du BoW pour trois valeurs alpha (0,2 , 0,4 et 0,6).

La deuxième méthode a surpassé l'approche prétraitée BoW pour deux valeurs alpha (0,1 et 0,6) et l'approche non prétraitée BoW pour toutes les valeurs alpha.

Dans la troisième méthode, les résultats des trois valeurs alpha (0,1, 0,2 et 0,6) étaient meilleurs par rapport à l'approche prétraitée du BoW, et le BoW non prétraité était le moins performant.

ID Méthode / Fonction Objectif	Méthode 1	Méthode 2	Méthode 3
$0.1 \cdot Taille + 0.9 \cdot Qualité$	0.7953	**0.9578**	**0.9914**
$0.2 \cdot Taille + 0.8 \cdot Qualité$	0.9319	0.9497	**0.9865**
$0.3 \cdot Taille + 0.7 \cdot Qualité$	0.8640	0.9171	0.9247
$0.4 \cdot Taille + 0.6 \cdot Qualité$	0.9190	0.9439	0.9706
$0.5 \cdot Taille + 0.5 \cdot Qualité$	0.8751	0.9361	0.9461
$0.6 \cdot Taille + 0.4 \cdot Qualité$	0.8741	**0.9534**	**0.9963**
$0.7 \cdot Taille + 0.3 \cdot Qualité$	0.8682	0.9461	0.9759
$0.8 \cdot Taille + 0.2 \cdot Qualité$	0.9274	0.9171	0.9247
$0.9 \cdot Taille + 0.1 \cdot Qualité$	0.8643	0.9439	0.9706
Sac de mots (prétraité)	0.9512		0.9770
Sac de mots (non prétraité)	0.8898		0.7920

TABLE 6.2: Résultats expérimentaux de l'approche proposée avec différentes fonctions objectives.

6.7 Conclusion

Dans ce chapitre, nous avons proposé une approche bio-inspirée pour faire correspondre des documents textuels hétérogènes, et cela pour évaluer notre principale contribution pour identifier de meilleurs descripteurs textuels. Dans la première étape, nous prétraitons les données en les nettoyant pour obtenir un vecteur de caractéristiques.

La deuxième phase met en évidence la sélection de l'ensemble minimal de caractéristiques qui véhicule la sémantique des documents textuels à l'aide de l'algorithme génétique d'inspiration quantique.

Sur la base des caractéristiques sélectionnées lors de la phase précédente, nous effectuons une mise en correspondance des documents. Afin de valider l'approche proposée, trois ensembles de documents provenant de sources différentes, sont utilisés pour récupérer leurs caractéristiques optimales. Ensuite, la mesure MRR est utilisée pour évaluer la précision d'appariement. L'approche proposée a surpassé l'approche classique BoW dans plusieurs configurations.

Chapitre 7

Impact de pré-traitement sur une approche bio-inspirée supervisée pour faire correspondre des documents texte

Sommaire

7.1 Introduction

Dans ce chapitre, nous proposons la version supervisée de l'approche bio-inspirée déjà proposée dans le chapitre précèdent, en l'évaluant l'impact des techniques de pré-traitements et sur les performances.

Comme la majorité des systèmes utilisant un espace vectoriel qui servent à l'évaluation de la similarité textuelle, le système que nous proposons est composé principalement de trois modules : pré-traitement, sélection de caractéristiques et mise en correspondance. Les systèmes sont différents dans la manière dont ces trois phases sont accomplies. Par rapport au système déjà proposé dans le chapitre précédent, notre contribution réside premièrement dans la phase de sélection des caractéristiques, où la fonction objectif a été modifiée, et deuxièmement au niveau du prétraitement, où nous avons étudié d'une façon détaillée l'impact des tâches de prétraitement sur la performance du système. Cette étude est réalisée sur les techniques de pré-traitement largement utilisées dans la littérature (voir section 2.2), y compris la suppression des nombres, la suppression des mots vides, la conversion en minuscules, la prise en compte des n-grammes, le stemming et la lemmatisation, de sorte que toutes les combinaisons possibles de ces tâches de pré-traitement sont considérées et comparées.

En résumé, nous cherchons à répondre aux questions suivantes : La phase de prétraitement est-elle cruciale pour le développement de meilleurs descripteurs textuels, afin de réaliser la tâche d'évaluation de similarité sémantique ? Si oui, quelle est la technique (combinaison de techniques) la plus appropriée ?

Le reste de ce chapitre est organisé comme suit : La section 7.2 représente la motivation de cette contribution. Une description du système d'évaluation de la similarité sémantique proposé est donnée en section 7.4. Dans la section 7.5, nous présentons les expérimentations et les résultats obtenus . Enfin, la conclusion et les travaux futurs sont indiqués dans la dernière section.

7.2 Motivation

L'étape de pré-traitement de texte représente une phase fondamentale de tout système de NLP, car les données textuelles en général contiennent des formats spéciaux tels que les nombres, les dates et les mots les plus courants connus par des mots vides tels que les prépositions, les articles et des pronoms qui ne sont pas pertinents pour aider le processus de fouille de texte. Ces données textuelles peuvent être éliminées afin de nettoyer le texte et le présenter au système dans un format normalisé.

Cette tâche de préparation des données affecte les performances des systèmes NLP en termes de temps d'exécution, de taille de mémoire et de précision du système. Dans la littérature du domaine de fouille de texte, nous trouvons plusieurs études traitant l'impact de l'étape de pré-traitement, en particulier la classification de texte [62, 158], la catégorisation des thèmes [31], l'analyse des sentiments [31, 52], la détection des thèmes [146] et l'identification de l'auteur [130]. Peu d'attention a été accordée dans la tâche de similarité sémantique textuelle qui est la tâche que nous traitons spécifiquement dans ce travail.

7.3 Travaux connexes

Dans les travaux présentés dans [158], les auteurs ont étudié l'impact des méthodes de pré-traitement largement utilisées sur la tâche de classification de textes s'appliquant à différents domaines et langues. L'examen a été réalisé en utilisant différentes combinaisons des tâches de pré-traitements en tenant compte de divers aspects tels que la précision, le domaine, la langue et la réduction des dimensions. L'étude a révélé que certaines combinaisons de tâches de pré-traitement en fonction du domaine et de la langue peuvent apporter une amélioration significative de la précision de la classification, tandis que certaines combinaisons inappropriées peuvent également dégrader la précision. Les auteurs dans [158] ont également noté qu'il existe des tâches de pré-traitements particulières telles que la conversion en minuscules qui peuvent améliorer la qualité de la classification en termes de précision et de réduction de dimension indépendamment du domaine et de la langue utilisés, et qu'il n'y a pas de combinaison parfaite des tâches de pré-traitement pouvant fournir des résultats de classification optimaux pour tous les domaines et langues étudiés. Les auteurs dans [158] ont également recommandé l'importance des mots vides, alors que dans la littérature, les chercheurs supposent généralement que les mots vides ne sont pas pertinents à considérer dans le processus. Par conséquent, pour appliquer la classification de textes sur tous les domaines et les langues, les utilisateurs devraient analyser soigneusement toutes les combinaisons possibles plutôt que d'en choisir ou d'en supprimer certaines afin de trouver la combinaison appropriée.

En ce qui concerne la catégorisation des sujets, les auteurs dans [31] ont déduit qu'une technique de tokenisation simple fonctionne en général mieux que des techniques de pré-traitement plus complexes telles que la lemmatisation ou la prise en compte des n-grammes. Ce type de conclusion ne se vérifie pas toujours dans les domaines spécialisés comme la santé [31].

Dans [52], une étude sur l'impact des stratégies de pré-traitement est réalisée pour la tâche de l'analyse des sentiments. Les résultats montrent que la combinaison entre le stemming et la suppression des mots vides affecte négativement les performances de la classification pour un ensemble de données et améliore légèrement la classification pour un autre ensemble de données. Ceci était justifié par le fait que les algorithmes de stemming utilisés sont des algorithmes simples fournis par le logiciel Rapidminer [1], et ces derniers ont des taux d'erreur élevés. Les résultats montrent également que les n-grammes de mots et de caractères ont également amélioré les résultats, car les n-grammes ont aidé à capturer les phrases négatives et les phrases courantes utilisées pour exprimer des sentiments.

L'identification de l'auteur est un autre domaine, où l'auteur dans [130] a examiné l'impact de la tâche de pré-traitement de texte turc. Plusieurs combinaisons de tâches de pré-traitement fondamentales ont été considérées. D'après les résultats de l'évaluation expérimentale, on peut relever que les caractéristiques du bigramme ne doivent pas être utilisées seules. L'utilisation d'unigramme ou de la combinaison de caractéristiques unigramme et bigramme garantirait des performances plus élevées pour l'identification des auteurs en turc. Même si les performances de classification les plus élevées pour chaque classificateur sont atteintes lorsque la suppression des mots vides est désactivée, conserver ces derniers à l'intérieur du texte aiderait également à obtenir de meilleures performances même si un algorithme de classification

1. https ://rapidminer.com/

différent est utilisé. Cependant, une étape de recherche de racine peut être nécessaire selon le classificateur utilisé pour l'identification de l'auteur. En bref, nous concluons qu'une combinaison appropriée de tâches de pré-traitements dépend du domaine et de la tâche à réaliser.

7.4 Éléments de contribution

Dans cette section, nous allons présenter les éléments de contribution dans nos travaux d'évaluation de l'impact du pré-traitement dans notre approche bio-inspirée.

Classifieur kNN

Un algorithme k-plus proche-voisin ou ce que l'on appelle en anglais k-Nearest Neighbors (kNN), est une méthode de classification des données qui a été proposée par Thomas Cover [10]. L'idée est que l'on utilise une grande quantité de données d'apprentissage, où chaque point de données est caractérisé par un ensemble de variables. Conceptuellement, chaque point est tracé dans un espace de grande dimension, où chaque axe de l'espace correspond à une variable individuelle. Lorsque nous avons un nouveau point de données (test), nous voulons trouver les K voisins les plus proches (c'est-à-dire les plus similaires). Le nombre K peut être choisi comme la racine carrée de N, le nombre total de points dans l'ensemble de données d'apprentissage, (ainsi, si N est 400, $K = 20$) [124]. Les principaux avantages du kNN pour la classification sont :

— Mise en œuvre très simple,

— Robuste vis-à-vis de l'espace de recherche ; par exemple, les classes n'ont pas besoin d'être séparables linéairement,

— Le classifieur peut être mis à jour en ligne à très peu de frais lorsque de nouvelles instances avec des classes connues sont présentées,

— Peu de paramètres à régler : métrique de distance et k.

7.5 Description global du système

Dans notre approche, la méthode suivie pour estimer la similarité sémantique entre les paires de textes passe par trois étapes comme le montre la figure 7.1, le pré-traitement qui transforme chaque document texte brut en un vecteur de caractéristiques sans bruit en utilisant la technique de pré-traitement appropriée. À partir de ces vecteurs, une deuxième phase sélectionne les caractéristiques les plus pertinentes en utilisant un QIGA qui génère des vecteurs plus réduits. La dernière étape est la correspondance de textes utilisant une mesure de similarité Cosinus.

7.5.1 Pré-traitement des données

Dans notre étude, nous avons appliqué chaque technique de pré-traitement séparément, puis nous avons combiné les différentes techniques de pré-traitement afin de révéler toutes les interactions possibles entre elles. Puisque les combinaisons combinant la lemmatisation et le stemming ont été éliminées en raison du même objectif

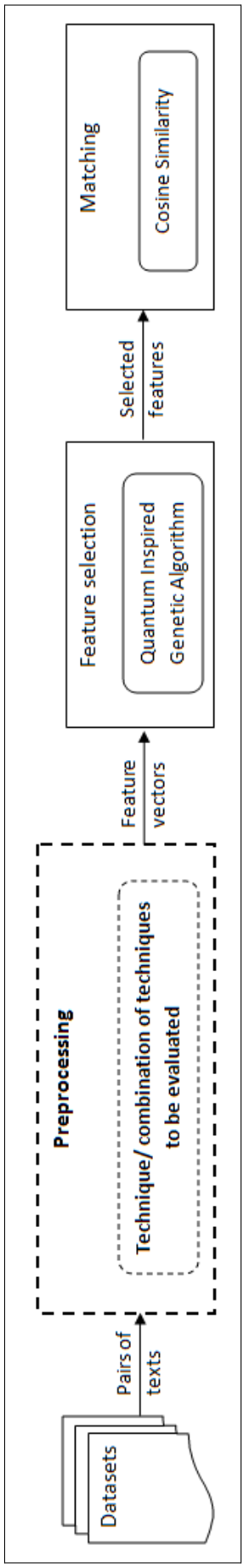

FIGURE 7.1: L'approche bio-inspirée supervisée proposée pour faire correspondre les documents textuels

visé par les deux derniers, 41 combinaisons différentes sont obtenues comme indiqué dans le tableau 7.1. Dans tous les cas, la ponctuation est ignorée et toujours considérée comme un séparateur des *tokens*.

Afin de faciliter la lisibilité, pour chaque combinaison, nous définissons avec une notation binaire si les nombres sont éliminés ou conservés dans le texte (T pour la suppression des nombres), si les termes sont convertis en minuscules ou conservés dans leur forme d'origine (L pour la conversion en minuscules), si les mots vides sont éliminés ou conservés dans le texte (S pour la suppression des mots vides), si les *tokens* sont mono-gramme ou mono/bi/tri-grammes (N pour n-grammes), si les termes sont réduits à leur forme «stem» ou «lemme» ou conservés dans leurs formes fléchies (M pour le stemming et Z pour la lemmatisation).

7.5.2 Sélection des caractéristiques

Les vecteurs de caractéristiques générés par l'étape précédente sont sur le point d'être réduits dans cette étape à l'aide de QIGA (voir section 6.4). Comme il a été décrit dans la section 6.2, cet algorithme offre plusieurs avantages dont le fait de fournir un bon équilibre entre l'exploration et l'exploitation dans l'espace de recherche, une complexité réduite et une énorme puissance de calcul. Le principe de fonctionnement d'un tel algorithme est d'explorer l'espace de différents sous-ensembles de l'ensemble de toutes les caractéristiques. La validité et la fiabilité des sous-ensembles de caractéristiques sélectionnés sont mesurées en appelant une fonction d'évaluation (ou fonction objectif) sur l'espace de caractéristiques réduit correspondant.

7.5.2.1 Fonction objectif

Afin d'évaluer la précision de l'algorithme, nous calculons à chaque itération la similarité entre les paires de textes à travers une similarité Cosinus entre leurs vecteurs TF-IDF, puis le classifieur kNN est utilisé pour classer les résultats obtenus. La valeur de fitness de l'algorithme est choisie comme étant le score F1.

$$F1 = \frac{2TP}{2TP + FP + FN} \tag{7.1}$$

où TP, FP et FN sont respectivement des nombres de vrais positifs, de faux positifs et de faux négatifs.

7.5.3 Évaluation de la mise en correspondance

Dans cette dernière étape, les meilleurs vecteurs de caractéristiques générés par l'étape précédente qui donne la correspondance la plus pertinente sont évalués avec des métriques standards, à savoir l'exactitude (*Accuracy*), la precision (*Precision*) et le rappel (*Recall*). Ceux-ci sont définis comme suit :

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{7.2}$$

$$Precision = \frac{TP}{TP + FP} \tag{7.3}$$

N	ID	Suppression des nombres (T)	Conversion en miniscules (L)	Suppression des mots vides (S)	N-grammes (N)	Stemming (M)	Lemmatisation (Z)
1	NZ	0	0	0	1	0	1
2	NM	0	0	0	1	1	0
3	SZ	0	0	1	0	0	1
4	SM	0	0	1	0	1	0
5	SN	0	0	1	1	0	0
6	SNZ	0	0	1	1	0	1
7	SNM	0	0	1	1	1	0
8	LZ	0	1	0	0	0	1
9	LM	0	1	0	0	1	0
10	LN	0	1	0	1	0	0
11	LNZ	0	1	0	1	0	1
12	LNM	0	1	0	1	1	0
13	LS	0	1	1	0	0	0
14	LSZ	0	1	1	0	0	1
15	LSM	0	1	1	0	1	0
16	LSN	0	1	1	1	0	0
17	lSNZ	0	1	1	1	0	1
18	lSNM	0	1	1	1	1	0
19	TZ	1	0	0	0	0	1
20	TM	1	0	0	0	1	0
21	TN	1	0	0	1	0	0
22	TNZ	1	0	0	1	0	1
23	tNM	1	0	0	1	1	0
24	TS	1	0	1	0	0	0
25	TSZ	1	0	1	0	0	1
26	TSM	1	0	1	0	1	0
27	TSN	1	0	1	1	0	0
28	TSNZ	1	0	1	1	0	1
29	TSNM	1	0	1	1	1	0
30	TL	1	1	0	0	0	0
31	TLZ	1	1	0	0	0	1
32	TLM	1	1	0	0	1	0
33	TLN	1	1	0	1	0	0
34	TLNZ	1	1	0	1	0	1
35	TLNM	1	1	0	1	1	0
36	TLS	1	1	1	0	0	0
37	TLSZ	1	1	1	0	0	1
38	TLSM	1	1	1	0	1	0
39	TLSN	1	1	1	1	0	0
40	TLSNZ	1	1	1	1	0	1
41	TLSNM	1	1	1	1	1	0

TABLE 7.1: Combinaisons de techniques de pré-traitement.

$$Recall = \frac{TP}{TP + FN} \tag{7.4}$$

où TP, FP et FN sont respectivement des nombres de vrais positifs, de faux positifs et de faux négatifs.

7.6 Étude expérimentale

Dans cette section, nous allons présenter l'étude expérimentale de l'application de notre approche sur la base de données MicroSoft Research Paraphrase Corpus [46].

7.6.1 Description du jeu de données

Afin de valider les résultats de nos expérimentations, nous avons besoin d'un ensemble de paires de textes sémantiquement liées et de paires qui ne sont pas liées. Nous avons donc utilisé MicroSoft Research Paraphrase Corpus (MSRPC) [46], MSRPC contient 5800 paires de phrases tirées de sources d'information sur le Web (4076 pour l'apprentissage et 1724 pour le test), ainsi que des annotations humaines indiquant si chaque paire capture une relation paraphrase / équivalence sémantique.

Paire de phrases	**Équivalence sémantique**
PCCW's chief operating officer, Mike Butcher, and Alex Arena, the chief financial officer, will report directly to Mr So. Current Chief Operating Officer Mike Butcher and Group Chief Financial Officer Alex Arena will report to So.	1
A tropical storm rapidly developed in the Gulf of Mexico Sunday and was expected to hit somewhere along the Texas or Louisiana coasts by Monday night. A tropical storm rapidly developed in the Gulf of Mexico on Sunday and could have hurricane-force winds when it hits land somewhere along the Louisiana coast Monday night.	0

TABLE 7.2: Exemple de paires de phrases de Microsoft Research Paraphrase Corpus

7.6.2 Spécification de paramètres

Dans cette étude, les valeurs des paramètres sont obtenues après avoir effectué plusieurs expérimentations sur chaque variable afin de trouver la meilleure valeur, la configuration suivante est utilisée comme réglage des paramètres de l'algorithme : la taille de la population a été choisie à 10, l'angle de rotation soit de $\pi/40$, les probabilités de mutation et de croisement soient respectivement de 0,1 et 0,9.

7.6.3 Résultats expérimentaux et discussion

Dans cette section, nous allons présenter les résultats de l'approche pour l'application de mono pré-traitement et multi pré-traitement.

7.6.3.1 Mono pré-traitement

Dans les premières expérimentations, chaque technique de pré-traitement est appliquée séparément, afin d'évaluer son impact par rapport au cas «Sans pré-traitement». Les résultats de l'analyse expérimentale avec les six techniques de pré-traitement sont illustrés dans le tableau 7.3. Ce dernier comprend les scores $F1$, *Accuracy*, *Precision*, *Recall*, les dimensions des caractéristiques *Dim* et le temps étendu *Time* pour exécuter la fonction objectif en une itération, où les valeurs en gras sont les meilleures et les soulignées sont les moins performants.

		F1 score	Recall	Precision	Accuracy	Time	Dim
	Sans pré-traitement	0.756	0.796	0.721	0.659	20.248	7186
pré-traitement	**Suppression des nombres**	0.753	0.809	0.705	0.648	**16.857**	6367
	Conversion en miniscules	0.761	0.811	0.718	0.662	19.518	5696
	Suppression des mots vides	0.760	0.838	0.695	0.648	18.635	4726
	N-grammes	**0.780**	**0.877**	0.703	**0.671**	24.812	87713
	Stemming	0.762	0.799	**0.729**	0.668	39.960	**4681**
	Lemmatisation	0.766	0.814	0.722	0.668	34.897	7190

TABLE 7.3: Résultats expérimentaux pour l'utilisation de chaque technique de pré-traitement séparément.

Compte tenu des six techniques de pré-traitement, le score F1 le plus élevé est celui de n-grammes (0,780), qui joue dans ce cas le rôle d'une technique d'enrichissement des données, contrairement à d'autres techniques comme la suppression des nombres qui a donné (0.753) un mauvais résultat même par rapport à la méthode 'Sans pré-traitement' (0.756), cela signifie que l'élimination des nombres conduit à une perte d'information. En ce qui concerne les techniques restantes, bien que la plupart d'entre elles aient surpassé la méthode 'Sans pré-traitement', elles peuvent également avoir causé une perte d'informations, que ce soit la conversion en miniscules qui ignore les majuscules (0,761), la suppression des mots vides qui ignore les mots vides (0.760), stemming qui ignore les suffixes (0.762), et lemmatisation qui ignore les formes fléchies du mot (0.766).

Comme il a été mentionné en haut, notre système se base sur une approche de sélection de caractéristiques. Par conséquent, la dimension des vecteurs des caractéristiques doit également être discutée. Comme déjà dit, la technique des n-grammes a amélioré la performance du système en enrichissant les données, ce qui a conduit à une expansion de la dimensionnalité des caractéristiques (87713), cela confirme que la sélection des caractéristiques ne conduit pas nécessairement à une réduction de ces derniers. D'autre part, la plus petite dimension des caractéristiques (qui est la meilleure) égale à 4681 est celle de la technique de stemming qui à son tour a donné le temps d'exécution le plus important.

Selon ces premières expérimentations, nous pouvons répondre à la première question "La phase de pré-traitement est-elle indispensable pour le développement de meilleurs descripteurs textuels, dans la tâche d'évaluation de similarité sémantique

basée sur QIGA ?" par l'affirmative, l'étape de pré-traitement est nécessaire pour améliorer la performance de notre système.

7.6.3.2 Multi pré-traitement

Pour répondre à la deuxième question "Quelle technique (combinaison de techniques) est la plus appropriée ?" nous avons évalué toutes les combinaisons possibles de ces techniques comme le montre la figure 7.2.

Les résultats montrent que plus de 95% des meilleurs résultats sont ceux obtenus à partir de techniques basées sur les n-grammes, telles que *'TSN'*, *'TLSN'* et *'SN'*. Nous remarquons également que les sept premières techniques sont caractérisées par l'existence de n-grammes et de la suppression des mots vides ensemble, cette combinaison permet au système d'extraire tous les termes simples et composés contenus dans l'ensemble de données ce qui conduit à son enrichissement sans être influencé par le bruit des mots vides.

Nous notons également sur la figure 7.2 que les techniques fondées sur le stemming ont donné des mauvais scores F1 et des petites tailles de vecteurs de caractéristiques, ce qui peut s'expliquer par la grande perte d'informations provoquée par cette technique. Les tailles des vecteurs des caractéristiques des techniques qui ont donné les meilleurs scores F1 sont généralement moyennes par rapport à toutes les tailles de caractéristiques obtenues. Ceci signifie qu'elles ont assuré un bon compromis entre l'enrichissement des vecteurs de caractéristiques et l'absence de bruit. Concernant, le temps d'exécution, les techniques de pré-traitement avec les meilleurs scores F1 ont consacré un temps d'exécution relativement réduit.

	Mono pré-traitement		**Multi pré-traitement**	
	N-grams (meilleur)	**la suppression des nombres (pire)**	**TSN (meilleur)**	**SM (pire)**
F1 score	0.780	0.753	**0.786**	0.749
Recall	0.877	0.809	**0.894**	0.829
Precision	0.703	**0.705**	0.702	0.684
Accuracy	0.671	0.648	**0.677**	0.631
Time (s)	24.812	16.857	16.443	**14.534**
Dim	87713	6367	58165	**5092**

TABLE 7.4: Comparaison des meilleurs et des pires résultats obtenus par des techniques de mono pré-traitement et de multi pré-traitement.

Le tableau 7.4 montre les meilleurs et les pires résultats obtenus à partir des techniques de mono pré-traitement et multi pré-traitement qui ont été expérimentées dans ce travail.

En comparant les meilleurs résultats obtenus par les techniques de mono pré-traitement et de multi pré-traitement, nous avons constaté que la technique *TSN* qui est une combinaison de la suppression des nombres, la suppression des mots vides et la prise en compte des n-grammes a surpassé la meilleure technique de mono pré-traitement qui est la prise en compte des n-grammes soit au score F1, à la dimension caractéristique ou au temps d'exécution. Cette technique a néanmoins

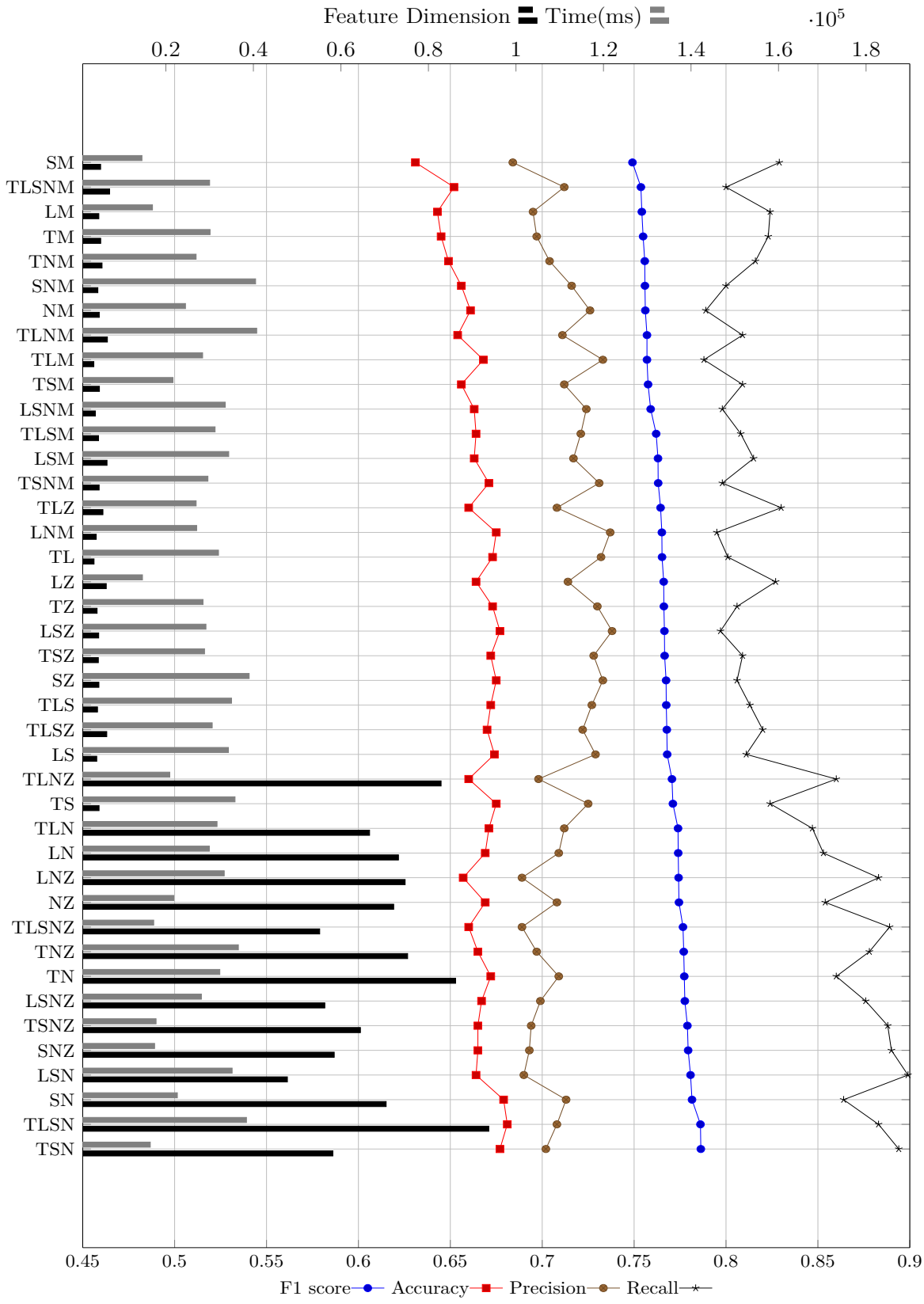

FIGURE 7.2: Résultats expérimentaux pour l'utilisation de toutes les combinaisons possibles de techniques de pré-traitement.

donné un vecteur de caractéristiques important et un temps d'exécution long, en raison de son effet d'enrichissement des données.

En ce qui concerne le vecteur de caractéristiques le plus court avec un temps d'exécution réduit, nous trouvons la technique *SM* qui a attaqué les données en supprimant les mots vides et en appliquant le stemming. Ceci se traduit par le pire score F1 obtenu, de même pour la technique de la suppression des nombres qui est basée sur le même principe. La technique *TSN* a donné le meilleur score F1 grâce au compromis obtenu entre enrichissement des données et élimination du bruit, ce qui répond à notre question "Quelle technique (combinaison de techniques) est la plus appropriée ?" par " La technique la plus appropriée est celle résultante de la combinaison de la suppression des nombres, la suppression des mots vides et la prise en compte des n-grammes.

7.7 Conclusion

Dans ce chapitre, nous avons examiné l'impact des techniques de pré-traitement largement utilisées sur la tâche de similarité sémantique en utilisant un algorithme génétique inspiré des quantiques afin d'améliorer la qualité des résultats. L'examen a été réalisé en utilisant chaque technique de pré-traitement séparément, à savoir la suppression des nombres, la suppression des mots vides, la conversion en miniscules, la prise en compte des n-grammes, stemming et lemmatisation. Enfin toutes les combinaisons possibles de ces techniques ont été examinées.

Les résultats expérimentaux ont montré qu'au niveau du mono pré-traitement, les techniques fondées sur l'élimination des données donnaient de pires résultats que la technique des n-grammes qui a enrichi les données. Cette technique a été renforcée au niveau du multi pré-traitement par une combinaison de la suppression des nombres et la suppression des mots vides qui a conduit à une amélioration des résultats.

Par conséquent, nous pouvons conclure que le pré-traitement des données est une étape indispensable dans notre système d'évaluation de similarité sémantique, et le résultat le plus intéressant de cette étude était l'importance de la technique des n-grammes pour le développement de meilleurs descripteurs textuels, en particulier lorsqu'elle est associée à des techniques de nettoyage en raison d'un bon compromis assuré entre l'enrichissement des données et l'élimination du bruit.

Chapitre 8

Étude de l'impact du pré-traitement sur la représentation distribuée

Sommaire

8.1 Introduction

L'étape la plus élémentaire pour la majorité des tâches de NLP est de convertir du texte en format numérique. Nous appelons cette étape, la représentation textuelle. Cette étape, bien qu'itérative, joue un rôle important dans le choix des caractéristiques des modèles et algorithmes d'apprentissage automatique. La représentation numérique de documents nous donne la possibilité d'effectuer des analyses significatives et crée également les instances sur lesquelles les algorithmes d'apprentissage automatique fonctionnent.

Dans les contributions que nous avons présenté dans les précédents chapitres, nous avons adopté une représentation discrète (voir section 3.2). Dans ce chapitre, nous allons adopter une autre méthode de représentation vectorielle dite distribuée. La représentation textuelle distribuée se produit lorsque la représentation d'un mot n'est pas indépendante ou mutuellement exclusive d'un autre mot et que leurs configurations représentent souvent diverses métriques et concepts dans une donnée. Ainsi les informations sur un mot sont distribuées le long du vecteur dans lequel il est représenté. Ceci est différent de la représentation discrète où chaque mot est considéré comme unique et indépendant l'un de l'autre (voir section 3.3).

Au lieu de ne représenter qu'un mot dans un vecteur, le plongement de documents (déjà décrit dans la section 3.3.2) transforme un document complet en un espace vectoriel, basé sur les vecteurs de ses mots constitutifs. Ces dernières années, un large éventail de méthodes de plongement a été proposé, leur capacité à convertir des documents en représentations vectorielles informatives laisse un impact positif sur un certain nombre d'applications, comme la classification de texte et l'analyse des sentiments [104], le classement image-phrase et la classification par type de question [92], les revues de produits et la classification de subjectivité [70], l'analyse des sentiments, la classification des documents ainsi que la relation sémantique [34].

Doc2vec (voir section 3.3.2.3) est une méthode qui modélise des séquences de texte telles que des phrases, des paragraphes ou des documents complets. Les vecteurs de plongement, dans cette méthode, sont générées pour les documents et les mots, et un mot cible est prédit via la concaténation de ses vecteurs de plongement dans un contexte de fenêtre glissante. Cette approche a surpassé les représentations de document précédentes, sur diverses tâches de compréhension de texte [40].

Les motivations de cette contribution sont présentées en section 8.2. La section 8.3 traite quelques travaux connexes. La section 8.4 donne une description de l'approche proposée. Les paramètres expérimentaux et les résultats sont traités dans la section 8.5. Enfin, nous fournissons des conclusions dans la section 8.6.

8.2 Motivation

Malgré ses avantages de permettre la généralisation à des documents plus longs, d'apprendre à partir de données non étiquetées, de prendre en compte l'ordre des mots, et selon Chen [34], l'approche Doc2vec souffre encore de deux limitations :

— le nombre de paramètres augmente avec la taille du corpus d'apprentissage, qui peut facilement atteindre des milliards,

— il est coûteux de générer des représentations vectorielles pour des documents invisibles au moment du test.

Cela a conduit à proposer une architecture de modèle améliorée, appelée Doc2vecC (voir section 3.3.2.4). Selon Chen [34], l'approche Doc2vecC a surpassé l'approche originale Doc2vec dans plusieurs tâches. Malgré cela, le plongement de documents est un sujet qui est encore activement exploré pour ses diverses caractéristiques, et parmi les points les moins éclairés figure l'effet des techniques de pré-traitement sur les approches récentes du plongement de documents.

Notre objectif dans ce chapitre vise à comparer les deux approches citées auparavant, Doc2vec et Doc2vecC, sous l'impact du pré-traitement morphosyntaxique. Nous visons ainsi à évaluer l'impact des techniques de pré-traitement de texte les plus connues : Nettoyage, stemming et lemmatisation sur les documents anglais sur la tâche de similarité sémantique.

8.3 Travaux connexes

Parmi les travaux de la littérature qui ont étudié l'impact des techniques de pré-traitement sur le plongement de mots, nous citons le travail présenté dans [31], qui évalue le rôle du pré-traitement de texte sur le plongement de mots sur les tâches de catégorisation de texte et d'analyse des sentiments. Les auteurs ont constaté qu'en général, une simple tokenisation donne des résultats égaux ou meilleurs par rapport aux techniques de pré-traitement plus complexes, telles que la lemmatisation ou le regroupement de plusieurs mots.

Cependant, Kiela et Clark [89] ont constaté que le stemming donne de meilleures performances globales que de ne pas réduire la granularité des caractéristiques. L'augmentation de la granularité des caractéristiques des contextes en incluant des catégories de grammaire catégorielle ou des balises POS n'offre aucune amélioration.

Lison et Kutuzov [107] ont présenté une analyse systématique du modèle Word2vec afin d'évaluer l'impact d'un ensemble d'aspects spécifiques, y compris la technique de suppression des mots vides. Dans ce travail, des modèles Skip-Gram continus ont été formés sur deux ensembles de données de langue anglaise et différentes combinaisons ont été évalués à la fois sur des tâches d'analogie et de similarité lexicale. Les expérimentations ont montré que la technique de suppression des mots vides n'influence pas vraiment les performances du modèle pour la tâche de similarité sémantique. Au contraire, la tâche d'analogie bénéficie substantiellement de cette technique de pré-traitement.

Dans [160], les auteurs discutent des modèles Word2vec, de leurs étapes de mise en œuvre et de leurs problèmes difficiles associés. Ils ont confirmé que plusieurs problèmes linguistiques de Word2vec peuvent être résolus lors de l'étape de pré-traitement. Par exemple, le traitement des mots homographes, où un mot peut être défini comme un verbe, un adverbe, une préposition ou un nom, est une problématique difficile. Il n'y a aucun moyen facile pour identifier ces mots identiques. La solution dans ce cas est de former un modèle à l'aide de textes prétraités par un étiqueteur morpho-syntaxique. Un autre problème difficile pour le plongement de mots est de déterminer si un mot particulier existe sous sa forme fléchie ; et une solution est d'appliquer la lemmatisation.

Dudchenko et Kopanitsa [51] ont développé un prototype de système d'extraction de données médicales, basé sur plusieurs architectures de réseaux de neurones artificiels, pour traiter des textes médicaux libres écrits en russe. Dans ce travail, la technique de pré-traitement de la lemmatisation s'est avérée avoir un bon impact

Travaux connexes	**Langue des données**	**Objectif de la tâche**	**Méthode de représentation**	**Techniques de pré-traitement**	**Technique(s) de pré-traitement qui donne de meilleurs résultats**
[89]	Anglais	* Similarité sémantique	Word2vec	* Lemmatisation * Stemming * POS-tagging * CCG-tagging	* Stemming
[31]	Anglais	* Catégorisation de texte * Analyse des sentiments	Word2vec	* Tokenisation * Conversion en minuscules * Lemmatisation * N-grammes	* Tokenisation
[107]	Anglais	* Similarité sémantique * Tâche d'analogie	Word2vec	* Suppression des mots vides	/
[20]	Arabe	* Analyse des sentiments	Doc2vec	* Cartographie de ponctuation * Stemming	* Stemming
[51]	Russe	* Extraction de données	Word2vec	* Lemmatisation	* Lemmatisation
[153]	Arabe	* Catégorisation	Word2vec	* Normalisation * Suppression des mots vides * Tokenisation	* Normalisation * Suppression des mots vides * Tokenisation
[165]	Anglais	* Similarité sémantique	Doc2vec	* Nettoyage * Stemming * Lemmatisation	* Stemming

TABLE 8.1: Différents travaux de l'état de l'art

sur les performances du classifieur, lorsqu'elle est appliquée avant la représentation Word2vec.

Une autre étude, celle de Soliman et al. [153], introduit un modèle pour améliorer la précision des groupes de documents arabes. Il a été constaté que la qualité des clusters de texte arabe a été considérablement améliorée en intégrant l'algorithme de catégorisation K-means avec Word2vec, où le pré-traitement du texte (la normalisation, la suppression des mots vides et la tokenisation) était une étape importante dans le système.

À notre connaissance, l'impact du pré-traitement sur le plongement de documents a rarement été étudié. Parmi les quelques travaux qui éclairent ce sujet, Barhoumi et al. [20] visaient à évaluer l'utilité d'une technique du plongement de documents, Doc2vec, dans un cadre d'analyse des sentiments en arabe. Les expériences menées dans cet article ont révélé la difficulté de traiter l'arabe par rapport à l'anglais. Pour cela, les auteurs ont testé certaines méthodes de pré-traitement, et les résultats ont montré qu'un stemming léger améliore les performances de classification. Le tableau 8.1 résume les travaux présentés ci-dessus.

8.4 Description globale de notre système

Dans notre approche, le processus suivi pour estimer la similarité sémantique entre les paires de texte passe par trois étapes comme le montre la figure 8.1. Il commence par un pré-traitement, qui transforme chaque document textuel brut en un texte sans bruit en utilisant les techniques de pré-traitement de texte mentionnées dans la section 2.2. Chaque technique est appliquée séparément, leurs combinaisons sont également prises en compte. Ensuite, les documents textuels résultants sont représentés via les deux approches de plongement de documents mentionnées dans la section 3.3. La dernière étape est la mise en correspondance de textes, en utilisant le classifieur kNN (voir la section 7.4), par rapport au corpus MSRPC (voir section 7.6.1).

8.5 Expérimentations

Dans cette section, nous allons présenter l'étude expérimentale de l'application de notre approche sur le corpus MSRPC.

8.5.1 Spécification de paramètres

Dans cette étude, plusieurs expérimentations ont été menées sur chaque paramètre Doc2vec afin de fixer leurs valeurs optimales. Les deux algorithmes d'apprentissage, PV-DM et PV-DBoW, sont appliqués ; et les valeurs suivantes sont utilisées pour toutes les expérimentations : $\alpha = 0,025$; $min\_\alpha = 0,025$; $vector\_size = 20$; $window\_size = 300$; $min\_count = 0$; $époque = 10$. Concernant Doc2vecC, les paramètres utilisés sont ceux inspirés de l'article original [34] : $vector\_size = 100$; $window\_size = 10$; $min\_count = 10$; $époque = 20$. Tous les hyper-paramètres réglés sont décrits dans le tableau 8.2.

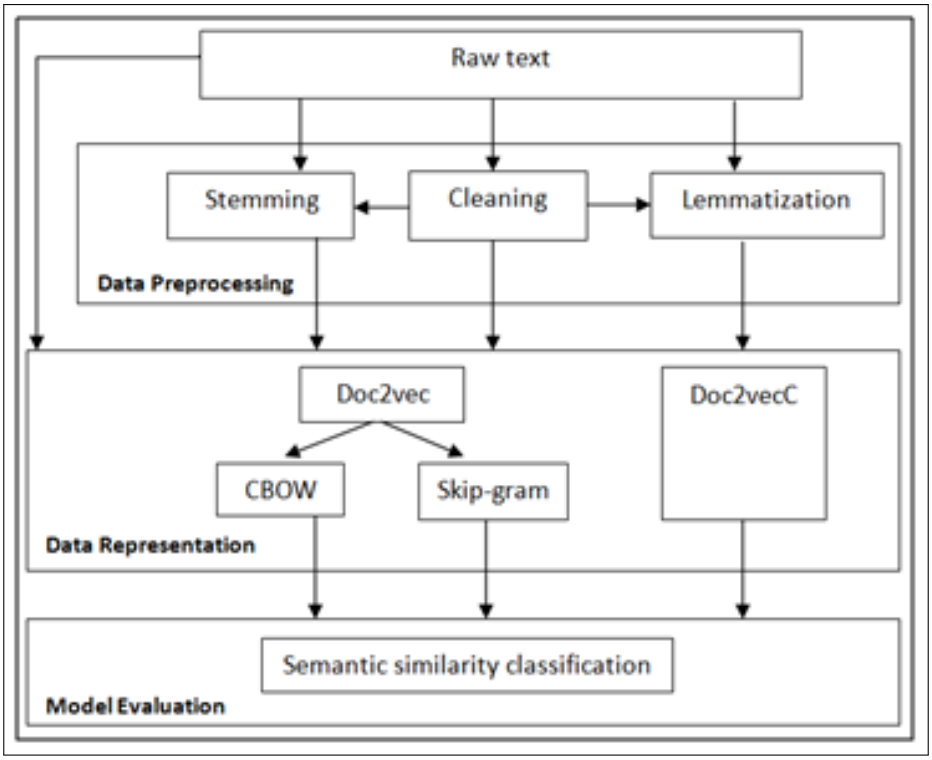

FIGURE 8.1: Synthèse de l'étude comparative de Doc2vec et Doc2vecC.

$dm(int1, 0)$	Définit l'algorithme d'apprentissage. Si dm = 1, la mémoire distribuée (PV-DM) est utilisée. Sinon, un sac distribué de mots (PV-DBOW) est utilisé.
$size(int)$	Dimensionnalité des vecteurs de caractéristiques.
$window(int)$	Distance maximale entre le mot actuel et le mot attendu dans une phrase.
$\alpha(float)$	Le taux d'apprentissage initial.
$min\_\alpha(float)$	Le taux d'apprentissage diminuera linéairement à min_α au fur et à mesure de la progression de l'apprentissage.
$min\_count(int)$	Ignore tous les mots dont la fréquence totale est inférieure à cela.
$epoch(int)$	Nombre d'itérations sur le corpus.

TABLE 8.2: Une description des hyper-paramètres Doc2vec et Doc2vecC

8.5.2 Résultats expérimentaux et discussion

Le tableau 8.3 montre les résultats expérimentaux obtenus en appliquant les techniques de pré-traitement sur les modèles Doc2vec et Doc2vecC. Dans cette étude, chaque technique de pré-traitement est appliquée séparément. Leurs combinaisons sont également envisagées, afin d'évaluer leur impact par rapport au cas «Sans prétraitement». Les résultats des 12 schémas résultants sont illustrés dans le tableau 8.3 qui comprend les taux de rappel, de précision et de F1. Les meilleures valeurs sont indiquées en gras et celles soulignées sont les moins performantes.

Le premier point fort de ce tableau est que l'algorithme PV-DM et le modèle

Doc2vecC fonctionnent tous les deux mieux que l'algorithme PV-DBoW dans tous les cas. Concernant les fragments de textes prétraités, nous observons qu'il a donné de meilleurs résultats par rapport au texte brut, ce qui signifie que le nettoyage, le stemming et la lemmatisation ont amélioré l'exactitude du classifieur en apportant plus de sémantique aux données. Les algorithmes de stemming ont eu un bon impact sur les modèles de plongement de documents en garantissant le meilleur compromis entre performance et précision. Il surpasse le nettoyage et la lemmatisation dans toutes les expériences, où *Lancaster stemmer* a donné le meilleur rappel, précision et score F1 pour l'algorithme PV-DM du modèle Doc2vec. *Snowball stemmer* a donné le meilleur rappel, précision et score F1 pour l'algorithme PV-DBoW du modèle Doc2vec.

Concernant le modèle Doc2vecC, *Porter Stemmer* a donné le meilleur rappel, et les meilleurs scores de précision et F1 ont été obtenus par *Lancaster Stemmer*. On remarque également que la combinaison de stemming et de lemmatisation avec les techniques de nettoyage n'a pas eu d'impact positif sur les modèles du plongement de documents. ceci signifie que l'élimination du bruit, dans notre cas, n'est pas une étape cruciale, contrairement à la technique de stemming, qui a fourni une amélioration encourageante.

			Sans pré-traitement	Avec pré-traitement												
				Nettoyage	Stemming			Stemming+Nettoyage			Lemmatization		Lemmatization+Nettoyage			
					Snowball	Porter	Lancaster	Snowball	Porter	Lancaster	Wordnet	Wordnet+POS Tagger	Wordnet	Wordnet+POS Tagger		
Doc2vec	PV-DM	Rec.	76.6	76.8	78.7	79.1	**79.3**	77.8	78.2	78.9	78.3	77.6	77.3	77.1		
		Prec.	69.3	69.5	69.3	69.8	**70.7**	69.1	68.6	69.5	69.4	69.7	70.2	69.3		
		F1	72.8	72.9	73.7	74.1	**74.8**	73.2	73.1	73.9	73.6	73.4	73.6	72.9		
	PV-DBoW	Rec.	74.9	73.4	**75.4**	75.1	74.2	75.2	75.1	73.4	75.1	74.7	74.5	75.0		
		Prec.	68.9	67.4	**69.0**	68.2	68.4	67.5	67.6	67.7	68.1	68.8	67.4	68.3		
		F1	71.8	70.2	**72.0**	71.5	71.1	71.1	71.1	70.4	71.4	71.6	70.7	71.5		
Doc2vecC		Rec.	76.3	75.9	78.9	**79.6**	79.2	76.5	74.6	76.6	78.2	76.7	77.4	77.2		
		Prec.	69.7	69.5	69.3	69.4	69.7	69.5	68.9	69.2	69.2	69.6	70.0	**70.3**		
		F1	72.8	72.5	73.9	74.1	**74.2**	72.8	71.5	72.7	73.4	72.9	73.6	73.7		

TABLE 8.3: Résultats expérimentaux de l'impact du pré-traitement sur les modèles 'Doc2vec' et 'Doc2vecC'

8.6 Conclusion

Dans ce chapitre, nous avons présenté une étude empirique sur l'impact du pré-traitement morphosyntaxique des données sur deux méthodes de plongement de documents : Vecteur de paragraphe (Doc2vec) et Vecteur de document à travers la corruption (Doc2vecC). Cela a été fait en appliquant les techniques de pré-traitement de texte les plus couramment utilisées, telles que le nettoyage, le stemming avec ses algorithmes les plus connus et la lemmatisation à l'aide de l'étiqueteur morpho-syntaxique de Wordnet. Différentes combinaisons de ces tâches de pré-traitement ont également été envisagées. Selon les résultats obtenus par ces expérimentations sur le corpus MSRPC, nous pouvons conclure que les techniques de pré-traitement, en particulier les algorithmes de stemming, améliorent la précision du classifieur.

Conclusion Générale

Au cours de cette thèse, nous avons proposé différentes contributions permettant d'extraire des descripteurs textuels optimaux pouvant être utilisés dans la tâche d'évaluation de similarité sémantique entre documents textuels. Notre objectif est d'élaborer des méthodes qui soient à la fois simples et efficaces donnant des résultats de bonne qualité dans un temps raisonnable. Ainsi, quatre approches ont été proposées dans cette thèse.

Dans un premier temps, nous avons proposé une approche fondée sur l'extraction de descripteurs linguistiques issus d'un texte et des termes propres à un thésaurus en appliquant une pondération sémantique spécifique. Notre méthode a tendance à rapprocher des textes ayant des thématiques proches ce qui permet de mettre en relation des données de qualité différente.

Une deuxième proposition était une approche non-supervisée fondée sur un algorithme bio-inspiré pour faire correspondre des documents textuels hétérogènes. Dans la première étape, nous prétraitons les données en les nettoyant pour obtenir un vecteur de caractéristiques (ou descripteurs) performant. La deuxième phase met en évidence la sélection de l'ensemble minimal de caractéristiques qui représente la sémantique des documents textuels à l'aide de l'algorithme génétique d'inspiration quantique QIGA. Sur la base des caractéristiques sélectionnées lors de la phase précédente, nous effectuons la mise en correspondance. Afin de valider l'approche proposée, trois ensembles de documents provenant de sources différentes, sémantiquement similaires, sont utilisés pour récupérer leurs caractéristiques optimales. Ensuite, la mesure MRR est utilisée pour évaluer la précision de mise en correspondance. L'approche proposée a surpassé des approches classiques. Les approches bio-inspirées, en particulier, QIGA pour la sélection des caractéristiques des données textuelles a eu un impact positif pour les tâches de similarité sémantique. Ce dernier a permis d'améliorer les performances du système proposé et il a également amélioré l'approche de base (sans sélection de caractéristiques). Cependant, notre approche peut démontrer ses limites au niveau de temps et de la complexité algorithmique. Plusieurs perspectives peuvent être proposées à ce travail. Premièrement, cette approche pourra être re-étudiée en utilisant les paradigmes de programmation parallèle afin de réduire son coût. Deuxièmement, ce travail pourra être validé en utilisant d'autres jeux de données. Troisièmement, nous songeons également à proposer d'autres méta-heuristiques pour le même problème ainsi que des hybridations entre les heuristiques et les méta-heuristiques pour traiter le problème d'évaluation de similarité sémantique textuelle.

Ensuite, nous avons continué avec une version supervisée de l'approche bio-inspirée déjà proposée, tout en examinant l'impact des techniques de pré-traitement largement utilisées pour la tâche de similarité sémantique. L'examen a été réalisé en utilisant chaque technique de pré-traitement séparément, à savoir la suppression

des nombres, la suppression des mots vides, la conversion en minuscules, la prise en compte des n-grammes, le stemming et la lemmatisation, puis toutes les combinaisons possibles de ces techniques. Par conséquent, nous pouvons conclure que le pré-traitement des données est une étape indispensable dans notre système d'évaluation de similarité sémantique. Le résultat le plus intéressant de cette étude était l'importance de la technique des n-grammes, en particulier lorsqu'elle est associée à des techniques de nettoyage en raison d'un bon compromis assuré entre l'enrichissement des données et l'élimination du bruit. Les techniques de pré-traitement, ont apporté un plus à l'évaluation de similarité sémantique. Ce dernier a permis d'améliorer les performances du système proposé et il a également surpassé l'approche de base (sans pré-traitement). Cependant, comme dans notre précédente étude, notre approche peut révéler des limites au niveau du temps et de la complexité algorithmique. Comme futurs travaux, nous avons l'intention d'appliquer d'autres variantes de l'algorithme génétique ou d'autres méta-heuristiques récentes dans le but de mieux explorer l'espace de recherche.

Nous avons enfin adopté des techniques de plongement de documents comme méthodes de représentation des données, tout en évaluant l'impact du pré-traitement sur ces méthodes. Une comparaison empirique est réalisée, en prenant la similarité sémantique comme étude de cas. Les deux méthodes de plongement de documents utilisées sont : Vecteur de paragraphe (Doc2vec) et Vecteur de document à travers la corruption (Doc2vecC). Cette étude a été réalisée en appliquant les techniques de pré-traitement de texte les plus connus : Le nettoyage, le stemming avec ses algorithmes les plus connus et la lemmatisation à l'aide de l'étiqueteur morpho-syntaxique de Wordnet. Différentes combinaisons de ces tâches de pré-traitement ont également été envisagées. Selon les résultats obtenus par nos expérimentations sur l'ensemble des données MSRPC, nous pouvons conclure que les techniques de pré-traitement, en particulier les algorithmes de stemming, améliorent l'exactitude du classifieur. Les méthodes de plongement de documents au niveau la de représentation des données ont apporté plus de sémantique, ce qui a amélioré l'évaluation de similarité textuelle. De plus, certaines techniques de pré-traitement ont aidé à améliorer les performances du système proposé pour les deux méthodes de représentation étudiées. Comme futurs travaux, nous avons l'intention d'appliquer d'autres méthodes de représentation, et réaliser des expérimentations à partir d'autres jeux de données.

Acronymes

ACO Ant Colony Optimization. 53
53, 54
54, 66
66, 67
67

ACOFS Ant Colony Optimization for Feature Selection. 66
66

AI Artificial Intelligence. 20
20, 21
21, 23
23

BoW Bag of Words. 38
38, 39
39, 65
65, 75
75, 77
77, 78
78

CBoW Continious Bag of Words. 40
40

DL Deep Learning. 22
22

ESA Explicit Semantic Analysis. 27
27

GA Genetic Algorithm. 52
52, 53
53, 65
65, 66
66, 67
67, 68
68, 70
70

GLSA Generalized Latent Semantic Analysis. 26
26

IR Information Retrieval. 25
25

kNN k-Nearest Neighbors. 82
82, 84
84, 95
95

LSA Latent Semantic Analysis. 26
26, 27
27

ML Machine Learning. 21
21, 22
22, 27
27

MRR Mean Reciprocal Rank. 70
70, 75
75, 77
77, 78
78, 100
100

MSRPC MicroSoft Research Paraphrase Corpus. 26
26, 27
27, 86
86, 95
95, 99
99, 101
101

NLP Natural Language Processing. 15
15, 23
23, 28
28, 30
30, 33
33, 37
37, 51
51, 52
52, 54
54, 58
58, 80
80, 92
92

PI Paraphrase Identification. 24
24

PMI-IR Pointwise Mutual Information - Information Retrieval. 27
27

POS Part Of Speech. 33
33, 44
44, 51
51, 93
93

PSO Particle Swarm Optimization. 67
67

PV-DBoW Paragraph Vector - Distributed Bag of Words. 43
43, 95
95, 97
97

PV-DM Paragraph Vector - Distributed Memory. 43
43, 95
95, 96
96, 97
97

QIEA Quantum Inspired Evolutionary Algorithm. 66
66

QIGA Quantum Inspired Genetic Algorithm. 68
68, 70
70, 73
73, 74
74, 76
76, 82
82, 84
84, 88
88, 100
100

RBF Radial Basis Function. 67
67

RR Reciprocal Rank. 70
70

SA Simulated Annealing. 51
51

STS Semantic Textual Similarity. 24
24, 44
44

TF-IDF Term Frequency – Inverse Document Frequency. 39
39, 74
74, 75
75, 77
77, 84
84

TF-KLD Term Frequency - Kullback Leibler Divergence. 27
27

TS Tabu Search. 51
51

TSP Travelling Salesman Problem. 53
53, 54
54

Production Scientifique

— **Communications dans des Conférences internationales**

— NourelhoudaYahi, Hacene Belhadef et Mathieu Roche. “Mise en correspondance de données textuelles hétérogènes à partir d’informations sémantiques”. In Atelier qualité des données du Web (QLOD’16) de la Conférence Internationale Francophone sur l’Extraction et la Gestion des Connaissances. Université de Reims Champagne-Ardenne. Reims, France, 2016, p. 1–6.

— NourelhoudaYahi et al. “Towards a bio-inspired approach to match heterogeneous documents”. In the 13th International Conference on Web Information Systems and Technologies. Porto, Portugal. Scitepress, 2017, p. 276–283.

— NourelhoudaYahi et Hacene Belhadef. “Morphosyntactic Preprocessing Impact on Document Embedding : An Empirical Study on Semantic Similarity”. In International Conference of Reliable Information and Communication Technology. Johor, Malaysia. Springer. 2019, p. 118–126.

— **Publication dans une revue**

— NourelhoudaYahi, Hacene Belhadef et Mathieu Roche. "Investigating the Impact of Preprocessing on Document Embedding : An Empirical Comparison". Accepted paper in : Int. J. Data Mining, Modelling and Management. (Inderscience), 2020.

Bibliographie

[1] Mohamed Abdel-Basset, Weiping Ding et Doaa El-Shahat. "A hybrid Harris Hawks optimization algorithm with simulated annealing for feature selection". In : *Artificial Intelligence Review* 54.1 (2021), p. 593-637.

[2] Charu C Aggarwal. *Machine learning for text*. Springer : Berlin/Heidelberg, Germany, 2018.

[3] Mehdi Hosseinzadeh Aghdam, Nasser Ghasem-Aghaee et Mohammad Ehsan Basiri. "Text feature selection using ant colony optimization". In : *Expert systems with applications* 36.3 (2009), p. 6843-6853.

[4] Eneko Agirre et al. "Ubc : Cubes for english semantic textual similarity and supervised approaches for interpretable sts". In : *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*. 2015, p. 178-183.

[5] RK Agrawal, Baljeet Kaur et Surbhi Sharma. "Quantum based whale optimization algorithm for wrapper feature selection". In : *Applied Soft Computing* 89 (2020), p. 106092.

[6] Siti Rohaidah Ahmad, Azuraliza Abu Bakar et Mohd Ridzwan Yaakub. "Ant colony optimization for text feature selection in sentiment analysis". In : *Intelligent Data Analysis* 23.1 (2019), p. 133-158.

[7] Ali Alessa et Miad Faezipour. "Tweet classification using sentiment analysis features and TF-IDF weighting for improved flu trend detection". In : *International Conference on Machine Learning and Data Mining in Pattern Recognition*. Springer. 2018, p. 174-186.

[8] Ammar Almomani, Mohammed Alweshah et Saleh Al. "Metaheuristic algorithms-based Feature Selection approach for Intrusion Detection". In : *Machine Learning for Computer and Cyber Security : Principle, Algorithms, and Practices* (2019), p. 184.

[9] Wojdan Alsaeedan, Mohamed El Bachir Menai et Saad Al-Ahmadi. "A hybrid genetic-ant colony optimization algorithm for the word sense disambiguation problem". In : *Information Sciences* 417 (2017), p. 20-38.

[10] Naomi S Altman. "An introduction to kernel and nearest-neighbor nonparametric regression". In : *The American Statistician* 46.3 (1992), p. 175-185.

[11] Mohammed Alweshah et Salwani Abdullah. "Hybridizing firefly algorithms with a probabilistic neural network for solving classification problems". In : *Applied Soft Computing* 35 (2015), p. 513-524.

[12] A. Amaral. "Paraphrase Identification and Applications in Finding Answers in FAQ Databases". In : 2013.

[13] Rie Kubota Ando. “Latent semantic space : Iterative scaling improves precision of inter-document similarity measurement”. In : *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*. 2000, p. 216-223.

[14] Ahmed Al-Ani. “Ant Colony Optimization for Feature Subset Selection.” In : *WEC (2)*. 2005, p. 35-38.

[15] Mohammad Javad Aranian, Moein Sarvaghad-Moghaddam et Monireh Houshmand. “Feature dimensionality reduction for recognition of Persian handwritten letters using a combination of quantum genetic algorithm and neural network”. In : *Majlesi Journal of Electrical Engineering* 11.2 (2017).

[16] Lourdes Araujo. “Genetic programming for natural language processing”. In : *Genetic Programming and Evolvable Machines* 21.1 (2020), p. 11-32.

[17] Oluleye H Babatunde et al. “A genetic algorithm-based feature selection”. In : *International Journal of Electronics Communication and Computer Engineering* 1.5 (2014), p. 889-905.

[18] Ricardo Baeza-Yates, Berthier Ribeiro-Neto et al. *Modern information retrieval*. T. 463. ACM press New York, 1999.

[19] Mohit Bansal, Kevin Gimpel et Karen Livescu. “Tailoring continuous word representations for dependency parsing”. In : *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2 : Short Papers)*. 2014, p. 809-815.

[20] Amira Barhoumi et al. “Document embeddings for Arabic sentiment analysis”. In : *Proceedings of the First Conference on Language Processing and Knowledge Management, LPKM 2017*. (Kerkennah (Sfax), Tunisia). Sept. 2017.

[21] Riza Batista-Navarro, Rafal Rak et Sophia Ananiadou. “Optimising chemical named entity recognition with pre-processing analytics, knowledge-rich features and heuristics”. In : *Journal of cheminformatics* 7.S1 (2015), S6.

[22] Ahmed Nasreddine Benaichouche. “Conception de métaheuristiques d’optimisation pour la segmentation d’images : application aux images IRM du cerveau et aux images de tomographie par émission de positons”. Thèse de doct. Université Paris-Est, 2014.

[23] Yoshua Bengio et al. “A neural probabilistic language model”. In : *Journal of machine learning research* 3.Feb (2003), p. 1137-1155.

[24] Rahul Bhagat et Eduard Hovy. “What is a paraphrase ?” In : *Computational Linguistics* 39.3 (2013), p. 463-472.

[25] William Blacoe et Mirella Lapata. “A comparison of vector-based representations for semantic composition”. In : *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*. 2012, p. 546-556.

[26] Ivan Boban, Alen Doko et Sven Gotovac. “Sentence Retrieval using Stemming and Lemmatization with Different Length of the Queries”. In : *Adv. Sci. Technol. Eng. Syst. J* 5 (2020), p. 349-354.

[27] Necva BÖLÜCÜ et Burcu CAN BUGLALILAR. "A cascaded unsupervised model for PoS tagging". In : *ACM Transactions on Asian and Low-Resource Language Information Processing* (2020).

[28] Ilhem BOUSSAID, Julien LEPAGNOT et Patrick SIARRY. "A survey on optimization metaheuristics". In : *Information sciences* 237 (2013), p. 82-117.

[29] Bart BROERE. "Syntactic properties of skip-thought vectors". Thèse de doct. Tilburg University, 2018.

[30] Davide BUSCALDI et al. "Irit : Textual similarity combining conceptual similarity with an n-gram comparison method". In : ** SEM 2012 : The First Joint Conference on Lexical and Computational Semantics–Volume 1 : Proceedings of the main conference and the shared task, and Volume 2 : Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*. 2012, p. 552-556.

[31] Jose CAMACHO-COLLADOS et Mohammad Taher PILEHVAR. "On the role of text preprocessing in neural network architectures : An evaluation study on text categorization and sentiment analysis". In : *arXiv preprint arXiv :1707.01780* (2017).

[32] Buyang CAO, Fred GLOVER et Cesar REGO. "A tabu search algorithm for cohesive clustering problems". In : *Journal of Heuristics* 21.4 (2015), p. 457-477.

[33] Daniel CER et al. "Semeval-2017 task 1 : Semantic textual similarity-multilingual and cross-lingual focused evaluation". In : *arXiv preprint arXiv :1708.00055* (2017).

[34] Minmin CHEN. "Efficient vector representation for documents through corruption". In : *arXiv preprint arXiv :1707.02377* (2017).

[35] Qingyu CHEN, Yifan PENG et Zhiyong LU. "BioSentVec : creating sentence embeddings for biomedical texts". In : *2019 IEEE International Conference on Healthcare Informatics (ICHI)*. IEEE. 2019, p. 1-5.

[36] Murphy CHOY. "Effective listings of function stop words for twitter". In : *arXiv preprint arXiv :1205.6396* (2012).

[37] Jim COWIE, Joe GUTHRIE et Louise GUTHRIE. "Lexical disambiguation using simulated annealing". In : *COLING 1992 Volume 1 : The 15th International Conference on Computational Linguistics*. 1992.

[38] Nick CRASWELL. "Mean Reciprocal Rank." In : *Encyclopedia of database systems* 1703 (2009).

[39] Zakaria Abd El Moiz DAHI, Chaker MEZIOUD et Amer DRAA. "A quantum-inspired genetic algorithm for solving the antenna positioning problem". In : *Swarm and Evolutionary Computation* 31 (2016), p. 24-63.

[40] Andrew M DAI, Christopher OLAH et Quoc V LE. "Document embedding with paragraph vectors". In : *arXiv preprint arXiv :1507.07998* (2015).

[41] Dipanjan DAS et Noah A SMITH. "Paraphrase identification as probabilistic quasi-synchronous recognition". In : *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*. 2009, p. 468-476.

[42] Dr. Dataman. *Looking into Natural Language Processing (NLP)*. 2018. URL : https://towardsdatascience.com/natural-language-processing-nlp-for-electronic-health-record-ehr-part-i-4cb1d4c2f24b. (accessed : 29.11.2020).

[43] Arthur Delbridge et John RL Bernard. *The macquarie dictionary*. Macquarie Library Sydney, 1981.

[44] Li Deng et Dong Yu. "Deep learning : methods and applications". In : *Foundations and trends in signal processing* 7.3–4 (2014), p. 197-387.

[45] Habib Dhahri et al. "Tabu search and machine-learning classification of benign and malignant proliferative breast lesions". In : *BioMed research international* 2020 (2020).

[46] William Dolan et al. "Unsupervised construction of large paraphrase corpora : Exploiting massively parallel news sources". In : *The 20th International Conference on Computational Linguistics*. 2004.

[47] Marco Dorigo, Mauro Birattari et Thomas Stutzle. "Ant colony optimization". In : *IEEE computational intelligence magazine* 1.4 (2006), p. 28-39.

[48] Marco Dorigo et Luca Maria Gambardella. "Ant colonies for the travelling salesman problem". In : *biosystems* 43.2 (1997), p. 73-81.

[49] Amer Draa et al. "A quantum-inspired differential evolution algorithm for solving the N-queens problem". In : *neural networks* 1.2 (2011).

[50] Haibin Duan. "Ant colony optimization : principle, convergence and application". In : *Handbook of Swarm Intelligence*. Springer, 2011, p. 373-388.

[51] Aleksei Dudchenko et Georgy Kopanitsa. "Comparison of Word Embeddings for Extraction from Medical Records". In : *International journal of environmental research and public health* 16.22 (2019), p. 4360.

[52] Rehab Duwairi et Mahmoud El-Orfali. "A study of the effects of preprocessing strategies on sentiment analysis for Arabic text". In : *Journal of Information Science* 40.4 (2014), p. 501-513.

[53] Fatima Zohra El Hlouli et al. "Detection of SMS Spam Using Machine-Learning Algorithms". In : *Embedded Systems and Artificial Intelligence*. Springer, 2020, p. 429-440.

[54] Tamer Elsayed, Jimmy Lin et Douglas W Oard. "Pairwise document similarity in large collections with MapReduce". In : *Proceedings of ACL-08 : HLT, Short Papers*. 2008, p. 265-268.

[55] E Feldina et O Makhnytkina. "Clustering Approach to Topic Modeling in Users Dialogue". In : *Proceedings of SAI Intelligent Systems Conference*. Springer. 2020, p. 611-617.

[56] Samuel Fernando et Mark Stevenson. "A semantic similarity approach to paraphrase detection". In : *Proceedings of the 11th annual research colloquium of the UK special interest group for computational linguistics*. 2008, p. 45-52.

[57] Evgeniy Gabrilovich, Shaul Markovitch et al. "Computing semantic relatedness using wikipedia-based explicit semantic analysis." In : *IJcAI*. T. 7. 2007, p. 1606-1611.

[58] Salton Gerard et J.McGill Michael. *Introduction to Modern Information Retrieval*. New York, NY, United States : McGraw-Hill Book Company, 1986. isbn : 978-0-07-054484-0.

[59] Ioannis Giagkiozis, Robin C Purshouse et Peter J Fleming. "An overview of population-based algorithms for multi-objective optimisation". In : *International Journal of Systems Science* 46.9 (2015), p. 1572-1599.

[60] Fred Glover. "Future paths for integer programming and links to artificial intelligence". In : *Computers operations research* 13.5 (1986), p. 533-549.

[61] David M Goldberg et Alan S Abrahams. "A Tabu search heuristic for smoke term curation in safety defect discovery". In : *Decision Support Systems* 105 (2018), p. 52-65.

[62] Teresa Gonçalves et Paulo Quaresma. "Evaluating preprocessing techniques in a text classification problem". In : *São Leopoldo, RS, Brasil : SBC-Sociedade Brasileira de Computação* (2005).

[63] F Grover et M Laguna. *Tabu search*. 1997.

[64] Lov K Grover. "A fast quantum mechanical algorithm for database search". In : *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*. 1996, p. 212-219.

[65] Vairaprakash Gurusamy et Subbu Kannan. "Performance analysis : Stemming algorithm for the english language". In : *International Journal for Scientific Research and Development* 5 (2017), p. 2321-613.

[66] Kuk-Hyun Han et Jong-Hwan Kim. "Genetic quantum algorithm and its application to combinatorial optimization problem". In : *Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No. 00TH8512)*. T. 2. IEEE. 2000, p. 1354-1360.

[67] MARSHALL HARGRAVE. *Deep Learning*. 2019. url : `https://www.investopedia.com/terms/d/deep-learning.asp`. (accessed : 11.10.2020).

[68] Samer Hassan. "Measuring semantic relatedness using salient encyclopedic concepts". Thèse de doct. University of North Texas, 2011.

[69] Darrall Henderson, Sheldon H Jacobson et Alan W Johnson. "The theory and practice of simulated annealing". In : *Handbook of metaheuristics*. Springer, 2003, p. 287-319.

[70] Felix Hill, Kyunghyun Cho et Anna Korhonen. "Learning distributed representations of sentences from unlabelled data". In : *arXiv preprint arXiv : 1602.03483* (2016).

[71] John Henry Holland et al. *Adaptation in natural and artificial systems : an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.

[72] Farzaneh Sajedi Hosseini et al. "Flash-flood hazard assessment using ensembles and Bayesian-based machine learning models : Application of the simulated annealing feature selection method". In : *Science of the total environment* 711 (2020), p. 135161.

[73] Farkhund Iqbal et al. “A hybrid framework for sentiment analysis using genetic algorithm based feature reduction”. In : *IEEE Access* 7 (2019), p. 14637-14652.

[74] MUHİTTİN IŞIK et HASAN DAĞ. “The impact of text preprocessing on the prediction of review ratings”. In : *Turkish Journal of Electrical Engineering & Computer Sciences* 28.3 (2020), p. 1405-1421.

[75] Fuad Issa et al. “Abstract meaning representation for paraphrase detection”. In : *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long Papers)*. 2018, p. 442-452.

[76] Young-Jae Jeon et Jae-Chul Kim. “Network reconfiguration in radial distribution system using simulated annealing and tabu search”. In : *2000 IEEE Power Engineering Society Winter Meeting. Conference Proceedings (Cat. No. 00CH37077)*. T. 4. IEEE. 2000, p. 2329-2333.

[77] Yangfeng Ji et Jacob Eisenstein. “Discriminative improvements to distributional sentence similarity”. In : *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. 2013, p. 891-896.

[78] Jay J Jiang et David W Conrath. “Semantic similarity based on corpus statistics and lexical taxonomy”. In : *arXiv preprint cmp-lg/9709008* (1997).

[79] G Johannsen, M Sini et G Salokhe. “Basic guidelines for managing AGROVOC”. In : *Agricultural Information and Knowledge Management Papers (FAO)* (2006).

[80] Karen Sparck Jones. “A statistical interpretation of term specificity and its application in retrieval”. In : *Journal of documentation* (1972).

[81] Laetitia Jourdan, Clarisse Dhaenens et El-Ghazali Talbi. “A genetic algorithm for feature selection in data-mining for genetics”. In : *Proceedings of the 4th Metaheuristics International ConferencePorto (MIC’2001)* (2001), p. 29-34.

[82] Md Monirul Kabir, Md Shahjahan et Kazuyuki Murase. “A new hybrid ant colony optimization algorithm for feature selection”. In : *Expert Systems with Applications* 39.3 (2012), p. 3747-3763.

[83] Janusz Kacprzyk et Witold Pedrycz. *Springer handbook of computational intelligence*. Springer, 2015.

[84] Subbu Kannan et Vairaprakash Gurusamy. “Preprocessing techniques for text mining”. In : *International Journal of Computer Science & Communication Networks* 5.1 (2014), p. 7-16.

[85] SAURAV KAUSHIK. *Introduction to Feature Selection methods with an example (or how to select the right variables ?)* 2016. url : https://www.analyticsvidhya.com/blog/2016/12/introduction-to-feature-selection-methods-with-an-example-or-how-to-select-the-right-variables/. (accessed : 20.10.2020).

[86] Tom Kenter, Alexey Borisov et Maarten De Rijke. “Siamese cbow : Optimizing word embeddings for sentence representations”. In : *arXiv preprint arXiv :1606.04640* (2016).

[87] Hamidreza Keshavarz et Mohammad Saniee Abadeh. “ALGA : Adaptive lexicon learning using genetic algorithm for sentiment analysis of microblogs”. In : *Knowledge-Based Systems* 122 (2017), p. 1-16.

[88] Aditya Khant et Mahendra Mehta. “Analysis of Financial News Using Natural Language Processing and Artificial Intelligence”. In : *1st International Conference on Business Innovation*. 2018, p. 176.

[89] Douwe Kiela et Stephen Clark. “A systematic study of semantic vector space model parameters”. In : *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC)*. 2014, p. 21-30.

[90] Adam Kilgarriff. “Thesauruses for natural language processing”. In : *International Conference on Natural Language Processing and Knowledge Engineering, 2003. Proceedings. 2003*. IEEE. 2003, p. 5-13.

[91] Adam Kilgarriff et Colin Yallop. “What’s in a Thesaurus ?” In : *LREC*. 2000, p. 1371-1379.

[92] Ryan Kiros et al. “Skip-thought vectors”. In : *Advances in neural information processing systems*. 2015, p. 3294-3302.

[93] Łukasz Kłyk et al. “Metaheuristics for tuning model parameters in two natural language processing applications”. In : *International Conference on Artificial Intelligence : Methodology, Systems, and Applications*. Springer. 2012, p. 32-37.

[94] Joost N Kok et al. “Artificial intelligence : definition, trends, techniques, and cases”. In : *Artificial intelligence* 1 (2009), p. 1-20.

[95] Tuomo Korenius et al. “Stemming and lemmatization in the clustering of finnish text documents”. In : *Proceedings of the thirteenth ACM international conference on Information and knowledge management*. 2004, p. 625-633.

[96] Zornitsa Kozareva et Andrés Montoyo. “Paraphrase identification on the basis of supervised machine learning techniques”. In : *International Conference on Natural Language Processing (in Finland)*. Springer. 2006, p. 524-533.

[97] Max Kuhn et Kjell Johnson. “An introduction to feature selection”. In : *Applied predictive modeling*. Springer, 2013, p. 487-519.

[98] Ankit Kumar et al. “Ask me anything : Dynamic memory networks for natural language processing”. In : *International conference on machine learning*. PMLR. 2016, p. 1378-1387.

[99] HM Keerthi Kumar et BS Harish. “Classification of short text using various preprocessing techniques : An empirical evaluation”. In : *Recent Findings in Intelligent Computing Techniques*. Springer, 2018, p. 19-30.

[100] Matt Kusner et al. “From word embeddings to document distances”. In : *International conference on machine learning*. PMLR. 2015, p. 957-966.

[101] Thomas K Landauer et Susan T Dumais. “A solution to Plato’s problem : The latent semantic analysis theory of acquisition, induction, and representation of knowledge.” In : *Psychological review* 104.2 (1997), p. 211.

[102] Thomas K Landauer, Peter W Foltz et Darrell Laham. “An introduction to latent semantic analysis”. In : *Discourse processes* 25.2-3 (1998), p. 259-284.

[103] Jey Han Lau et Timothy Baldwin. “An empirical evaluation of doc2vec with practical insights into document embedding generation”. In : *arXiv preprint arXiv :1607.05368* (2016).

[104] Quoc Le et Tomas Mikolov. “Distributed representations of sentences and documents”. In : *International conference on machine learning*. 2014, p. 1188-1196.

[105] Kenton Lee et al. “End-to-end neural coreference resolution”. In : *arXiv preprint arXiv :1707.07045* (2017).

[106] Elizabeth D Liddy. “Natural language processing”. In : *2nd edn. Encyclopedia of Library and Information Science* (2001).

[107] Pierre Lison et Andrey Kutuzov. “Redefining context windows for word embedding models : An experimental study”. In : *arXiv preprint arXiv :1704.05781* (2017).

[108] Xianggen Liu et al. “Unsupervised paraphrasing by simulated annealing”. In : *arXiv preprint arXiv :1909.03588* (2019).

[109] Victor SY Lo. “The true lift model : a novel data mining approach to response modeling in database marketing”. In : *ACM SIGKDD Explorations Newsletter* 4.2 (2002), p. 78-86.

[110] Julien Longhi et al. “Polititweets, corpus de tweets provenant de comptes politiques influents”. In : *Banque de corpus CoMeRe. Ortolang. fr* (2014).

[111] Majdi M Mafarja et Seyedali Mirjalili. “Hybrid whale optimization algorithm with simulated annealing for feature selection”. In : *Neurocomputing* 260 (2017), p. 302-312.

[112] Ana G Maguitman et al. “Algorithmic detection of semantic similarity”. In : *Proceedings of the 14th international conference on World Wide Web*. 2005, p. 107-116.

[113] Goutam Majumder et al. “Semantic textual similarity methods, tools, and applications : A survey”. In : *Computación y Sistemas* 20.4 (2016), p. 647-665.

[114] Nicholas Metropolis et al. “Equation of state calculations by fast computing machines”. In : *The journal of chemical physics* 21.6 (1953), p. 1087-1092.

[115] Tomas Mikolov et al. “Efficient estimation of word representations in vector space”. In : *arXiv preprint arXiv :1301.3781* (2013).

[116] Tomáš Mikolov, Wen-tau Yih et Geoffrey Zweig. “Linguistic regularities in continuous space word representations”. In : *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics : Human language technologies*. 2013, p. 746-751.

[117] George A Miller. *WordNet : An electronic lexical database*. MIT press, 1998.

[118] George A Miller et al. “Introduction to WordNet : An on-line lexical database”. In : *International journal of lexicography* 3.4 (1990), p. 235-244.

[119] David Milward. *What is Text Mining, Text Analytics and Natural Language Processing ?* 2019. URL : https://www.linguamatics.com/what-text-mining-text-analytics-and-natural-language-processing. (accessed : 20.11.2020).

[120] Melanie Mitchell. *An introduction to genetic algorithms*. MIT press, 1998.

[121] Muhidin Mohamed et Mourad Oussalah. "A hybrid approach for paraphrase identification based on knowledge-enriched semantic heuristics". In : *Language Resources and Evaluation* 54.2 (2020), p. 457-485.

[122] Luis Carlos Molina, Lluis Belanche et Àngela Nebot. "Feature selection algorithms : A survey and experimental evaluation". In : *2002 IEEE International Conference on Data Mining, 2002. Proceedings.* IEEE. 2002, p. 306-313.

[123] Megan A Moreno et al. "Applying natural language processing to evaluate news media coverage of bullying and cyberbullying". In : *Prevention science* 20.8 (2019), p. 1274-1283.

[124] P Nadkarni. "Core Technologies : Data Mining and "Big Data"". In : *Clinical Research Computing : A Practitioner's Handbook* (2016), p. 187-204.

[125] Ajit Narayanan et Mark Moore. "Quantum-inspired genetic algorithms". In : *Proceedings of IEEE international conference on evolutionary computation*. IEEE. 1996, p. 61-66.

[126] Sergei Nirenburg, Marjorie McShane et Stephen Beale. "Resolving paraphrases to support modeling language perception in an intelligent agent". In : *Semantics in Text Processing. STEP 2008 Conference Proceedings*. 2008, p. 179-192.

[127] Idowu O Oduntan et al. "A multilevel tabu search algorithm for the feature selection problem in biomedical data". In : *Computers & Mathematics with Applications* 55.5 (2008), p. 1019-1033.

[128] Luiz S Oliveira et al. "A methodology for feature selection using multiobjective genetic algorithms for handwritten digit string recognition". In : *International Journal of Pattern Recognition and Artificial Intelligence* 17.06 (2003), p. 903-929.

[129] Chris D Paice. "Another stemmer". In : *ACM Sigir Forum*. T. 24. 3. ACM New York, NY, USA. 1990, p. 56-61.

[130] Muhammet Yasin PAK et Serkan Günal. "THE IMPACT OF TEXT REPRESENTATION AND PREPROCESSING ON AUTHOR IDENTIFICATION." In : *Anadolu University of Sciences & Technology-A : Applied Sciences & Engineering* 18.1 (2017).

[131] Shay Palachy. *Document Embedding Techniques*. 2020. URL : https://www.topbots.com/document-embedding-techniques/. (accessed : 03.12.2020).

[132] David D Palmer. "Tokenisation and sentence segmentation". In : *Handbook of natural language processing* (2000), p. 11-35.

[133] Martin Porter. *Snowball : 'A Language For Stemming Algorithms'*. October 2001 (accessed September 20, 2019). URL : http://snowball.tartarus.org/texts/introduction.html.

[134] Martin F Porter et al. “An algorithm for suffix stripping.” In : *Program* 14.3 (1980), p. 130-137.

[135] Kedar Potdar, Taher S Pardawala et Chinmay D Pai. “A comparative study of categorical variable encoding techniques for neural network classifiers”. In : *International journal of computer applications* 175.4 (2017), p. 7-9.

[136] A. C. Ramos et M. Vellasco. “Quantum-inspired Evolutionary Algorithm for Feature Selection in Motor Imagery EEG Classification”. In : *2018 IEEE Congress on Evolutionary Computation (CEC)*. 2018, p. 1-8.

[137] Simon Réhel. “Catégorisation automatique de textes et cooccurrence de mots provenant de documents non étiquetés”. Mém. de mast. QUÉBEC : Université LAVAL, 2005.

[138] Philip Resnik. “Using information content to evaluate semantic similarity in a taxonomy”. In : *arXiv preprint cmp-lg/9511007* (1995).

[139] Philip Resnik. “Wordnet and distributional analysis : A class-based approach to lexical discovery”. In : *AAAI workshop on statistically-based natural language processing techniques*. 1992, p. 56-64.

[140] KA Ross et al. *Cross-validation. encyclopedia of database systems*. 2009.

[141] Margaret Rouse. *computational linguistics (CL)*. 2018. url : `https://searchenterpriseai.techtarget.com/definition/computational-linguistics-CL`. (accessed : 23.11.2020).

[142] CHRISTIAAN Royer. “Term representation with generalized latent semantic analysis”. In : *Recent advances in natural language processing IV : selected papers from RANLP* 292 (2005), p. 45.

[143] Gerard Salton et Christopher Buckley. “Term-weighting approaches in automatic text retrieval”. In : *Information processing & management* 24.5 (1988), p. 513-523.

[144] Miguel A Sanchez-Perez, Alexander Gelbukh et Grigori Sidorov. “Adaptive algorithm for plagiarism detection : The best-performing approach at PAN 2014 text alignment competition”. In : *International Conference of the Cross-Language Evaluation Forum for European Languages*. Springer. 2015, p. 402-413.

[145] Jürgen Schmidhuber. “Deep learning in neural networks : An overview”. In : *Neural networks* 61 (2015), p. 85-117.

[146] Roman Sergienko, Muhammad Shan et Alexander Schmitt. “A comparative study of text preprocessing techniques for natural language call routing”. In : *Dialogues with Social Robots*. Springer, 2017, p. 23-37.

[147] Peter W Shor. “Algorithms for quantum computation : discrete logarithms and factoring”. In : *Proceedings 35th annual symposium on foundations of computer science*. Ieee. 1994, p. 124-134.

[148] Wojciech Siedlecki et Jack Sklansky. “A note on genetic algorithms for large-scale feature selection”. In : *Handbook of pattern recognition and computer vision*. World Scientific, 1993, p. 88-107.

[149] Herbert A Simon. “Why should machines learn ?” In : *Machine learning*. Elsevier, 1983, p. 25-37.

[150] Amit Singhal et al. “Modern information retrieval : A brief overview”. In : *IEEE Data Eng. Bull.* 24.4 (2001), p. 35-43.

[151] Noah A Smith et Jason Eisner. “Annealing techniques for unsupervised statistical language learning”. In : *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*. 2004, p. 486-493.

[152] Richard Socher et al. “Dynamic pooling and unfolding recursive autoencoders for paraphrase detection”. In : *Advances in neural information processing systems*. 2011, p. 801-809.

[153] Hasnaa R. H. Soliman, Mohamed Grida et Mohamed Hassan. “Arabic Text Clustering based on K-means Algorithm with Semantic Word Embedding”. In : *Journal of Theoretical and Applied Information Technology* 97.21 (2019), p. 2498-2509.

[154] Kai Sheng Tai, Richard Socher et Christopher D Manning. “Improved semantic representations from tree-structured long short-term memory networks”. In : *arXiv preprint arXiv :1503.00075* (2015).

[155] Hichem Talbi, Amer Draa et Mohamed Batouche. “A novel quantum-inspired evolutionary algorithm for multi-sensor image registration”. In : *The International Arab Journal of Information Technology* 3.1 (2006), p. 9-15.

[156] Expert System Team. *What is Machine Learning ? A Definition*. 2020. url : `https://www.expert.ai/blog/machine-learning-definition/`. (accessed : 01.12.2020).

[157] Damien Teney, Lingqiao Liu et Anton van Den Hengel. “Graph-structured representations for visual question answering”. In : *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, p. 1-9.

[158] Alper Kursat Uysal et Serkan Gunal. “The impact of preprocessing on text classification”. In : *Information Processing & Management* 50.1 (2014), p. 104-112.

[159] Sumithra Velupillai et al. “Using clinical natural language processing for health outcomes research : overview and actionable suggestions for future advances”. In : *Journal of biomedical informatics* 88 (2018), p. 11-19.

[160] P Verma et B Khandelwal. “Word embeddings and its application in deep learning”. In : *International Journal of Innovative Technology and Exploring Engineering* 8.11 (2019), p. 337-341.

[161] Tanu Verma, R Renu et D Gaur. “Tokenization and filtering process in RapidMiner”. In : *International Journal of Applied Information Systems* 7.2 (2014), p. 16-18.

[162] Michael D Vose. *The simple genetic algorithm : foundations and theory*. MIT press, 1999.

[163] Yue Wang et al. “Topic model based text similarity measure for Chinese judgment document”. In : *International Conference of Pioneering Computer Scientists, Engineers and Educators*. Springer. 2017, p. 42-54.

[164] Bing Xue, Mengjie Zhang et Will N Browne. “Multi-objective particle swarm optimisation (PSO) for feature selection”. In : *Proceedings of the 14th annual conference on Genetic and evolutionary computation*. 2012, p. 81-88.

[165] Nourelhouda Yahi et Hacene Belhadef. “Morphosyntactic Preprocessing Impact on Document Embedding : An Empirical Study on Semantic Similarity”. In : *International Conference of Reliable Information and Communication Technology*. Springer. 2019, p. 118-126.

[166] Jihoon Yang et Vasant Honavar. “Feature subset selection using a genetic algorithm”. In : *Feature extraction, construction and selection*. Springer, 1998, p. 117-136.

[167] Silvia Casado Yusta. “Different metaheuristic strategies to solve the feature selection problem”. In : *Pattern Recognition Letters* 30.5 (2009), p. 525-534.

[168] Bilal M Zahran et Ghassan Kanaan. “Text feature selection using particle swarm optimization algorithm”. In : *World Applied Sciences Journal (Special Issue of Computer & IT)* 7 (2009), p. 69-74.

[169] Hongbin Zhang et Guangyu Sun. “Feature selection using tabu search method”. In : *Pattern recognition* 35.3 (2002), p. 701-711.

[170] Shichao Zhang, Chengqi Zhang et Qiang Yang. “Data preparation for data mining”. In : *Applied artificial intelligence* 17.5-6 (2003), p. 375-381.

[171] Justin Zobel et Alistair Moffat. “Exploring the similarity space”. In : *Acm Sigir Forum*. T. 32. 1. ACM New York, NY, USA. 1998, p. 18-34.

[172] Jaime Zornoza. *An Introduction to Feature Selection*. 2020. url : `https://towardsdatascience.com/an-introduction-to-feature-selection-dd72535ecf2b`. (accessed : 23.10.2020).