REPUBLIQUE ALGERIENNE
DEMOCRATIQUE ET POPULAIRE

Université Mohamed Khider – BISKRA

Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie

Département d'informatique

Mémoire

Présenté pour obtenir le diplôme de master académique en

Informatique

Parcours : Systèmes d'information, Optimisation et décision

Extraction les concepts textuels en utilisant la technique de deep learning

Préparer par:
Fatma Zahra Amrani

Suivi par :
DR: Belkacem Abdeli

Dédicaces

À

mes parents,

mon mari

mes frères et ma sœur,

toute la famille,

et mes amis,

je dédie ce modeste travail.

Remerciements

En tout premier lieu, je tiens à remercier Dieu le tout puissant de m'avoir donné le courage, et l'énergie pour terminer ce travail.

Je remercie du fond du cœur et avec un grand amour mes parents, mon mari, ma sœur et mes frères qui n'ont jamais cessé de croire en moi pendant toutes mes années d'études.

J'ai l'honneur et le plaisir de présenter ma profonde gratitude et mes sincères remerciements à mon encadreur **Dr. Abdelli Belkacem**, pour ses conseils précieux, ces orientations et le temps qu'il m'a accordé pour mon encadrement.

Mes remerciements vont également aux membres de jury pour m'avoir honoré par leur évaluation de ce travail.

Merci à tous.

Résumé/Abstract

Résumé:

Le texte est important car il contient une énorme quantité d'informations dont nous pouvons bénéficier en l'analysant et en extrayant ses concepts les plus importants.
Il est devenu nécessaire le développement d'outils automatisés permettant de traiter et d'analyser le texte de documents afin d'extraire ce qui est plus significatif, et de donner une vision générale sur le contenu.

Dans notre mémoire on s'intéresse à l'analyse et le traitement de documents afin d'extraire le contenu le plus significatif qui représente et décrire au mieux le document.

Dans notre travail, nous utilisons une approche **'Deep Learning'** pour repérer les concepts les plus représentatifs de texte.

Mot clés : texte Extraction de concepts simples et composés, Deep Learning, analyse syntaxique, Analyse sémantique.

Abstract:

The text is important because it contains a huge amount of information that we can benefit from by analyzing it and extracting its most important concepts. It has become necessary to develop automated tools to process and analyze text from documents in order to extract what is more meaningful, and to give a general view of the content.

In our thesis we are interested in the analysis and processing of documents in order to extract the most meaningful content that best represents and describes the document.

In our work, we use a "deep learning" approach to identify the most representative concepts of text.

Keywords: Text, Extraction of simple and compound concepts, Deep Learning, syntax analysis, Semantic analysis.

الملخص:

النص مهم لأنه يحتوي على كمية هائلة من المعلومات يمكننا الاستفادة منها بتحليله واستخراج أهم مفاهيمه. أصبح من الضروري تطوير أدوات آلية لمعالجة وتحليل النص من المستندات من أجل استخراج ما هو أكثر فائدة ، وإعطاء نظرة عامة للمحتوى.

نحن مهتمون في أطروحتنا بتحليل المستندات ومعالجتها من أجل استخراج المحتوى الأكثر أهمية الذي يمثل المستند ويصفه على أفضل وجه.

في عملنا ، نستخدم نهج "التعلم العميق" لتحديد المفاهيم الأكثر تمثيلاً للنص
الكلمات الرئيسية: نص ، استخلاص المفاهيم البسيطة والمركبة ، التعلم العميق ، تحليل النحو ، التحليل الدلالي

TABLE DE MATIERES

LISTE DES FIGURES

Chapitre1 :

Chapitre2 :

Chapitre03 :

Chapitre04 :

LISTE DES TABLEAUX

Chapitre 2 :

INTRODUCTION GENERALE

1. Contexte :

Au cours des dernières années, Les textes ont connu une grande importance grâce à la diffusion du monde virtuel et l'émergence des réseaux sociaux, des pages, des sites n'est pas sans une énorme quantité de textes, nous devons donc bien comprendre leur contenu et essayer d'en extraire les concepts de base les plus importants.

La grande masse de documents textuels disponibles actuellement ainsi que ses exploitations croissantes, nécessite le développement de méthodes et d'outils pour le traitement et l'exploration automatique du texte afin d'extraire le contenu le plus significatif.

Dans les temps anciens, le texte est traité et analysé pour extraire manuellement les concepts les plus importants, avec l'avènement de la modernisation, l'analyse de texte est devenue automatique, utilisant des techniques classiques qui prennent du temps et des efforts et donnent des résultats insatisfaisants.

Nouvellement, L'émergence d'une nouvelle technique 'Deep Learning', qui a donné des résultats très satisfaisants. Deep Learning (L'apprentissage profond) est un ensemble de techniques d'apprentissage automatique qui a permis des avancées importantes en intelligence artificielle ces dernières années. Dans l'apprentissage automatique, un programme analyse les données afin de tirer des règles qui permettront de tirer des conclusions.

Notre travail se situe dans le contexte de l'extraction des concepts les plus représentatifs à partir d'un ou plusieurs documents en utilisant les techniques de Deep Learning.

2. Motivation :

Les techniques classiques de l'extraction et l'analyse à partir du texte de documents considèrent le contenu d'un document comme une collection de mots clés indépendants.

Généralement, la seule information utilisée concernant les mots d'un document est leurs fréquences d'apparition dans la collection des documents. Ceci ne fournit aucune significations, ni sur les sens des termes, ni sur les relations sémantiques entre eux. Par exemple, il est impossible d'identifier que le terme «automobile » a le même sens du terme « voiture », et le sens du terme « bus »a une relation de correspondance sémantique avec le sens du terme « taxi », aussi il est impossible de détecter qu'un terme est le synonyme (ou l'hyperonyme) d'un autre terme.

Dans notre travail, nous avons utilisé l'extraction des concepts les plus significatifs en utilisant une nouvelle technique d'intelligence artificielle 'Deep Learning', afin de répondre aux questions suivantes :

Quelles informations devraient être extraites et comment cela est-il fait ?

Comment pouvons-nous extraire les concepts (sens) des termes dans un document ?

Comment détecter les relations sémantiques (sens) entre termes afin de repérer les meilleures significations à partir du contexte ?

3 ; Objectifs :

L'objectif de notre travail est: **Comment extraire des concepts textuels à l'aide de la technique d'apprentissage en profondeur**.

Nous allons d'abord essayer d'extraire et de maintenir les concepts complexes avec un processus d'analyse de mots. Ensuite, nous analyserons la grammaire et extrairons la signification et la nature des concepts.

Enfin, nous devons effectuer une analyse sémantique et calculer la similitude sémantique entre les documents.

4. Organisation du mémoire :

Le présent mémoire s'articule autour de quatre (04) chapitres :

Chapitre I : Analyse Automatique De Texte :

Dans ce chapitre, nous présentons un état de l'art sur l'extraction des concepts dans les documents non-structurés. Tout d'abord, nous présentons le principe de l'extraction des termes et des concepts ainsi que le principe d'analyse de documents textuels, puis nous détaillons le principe de l'analyse syntaxique et sémantique de documents non-structurés.

Chapitre II : Deep Learning Pour Le Texte :

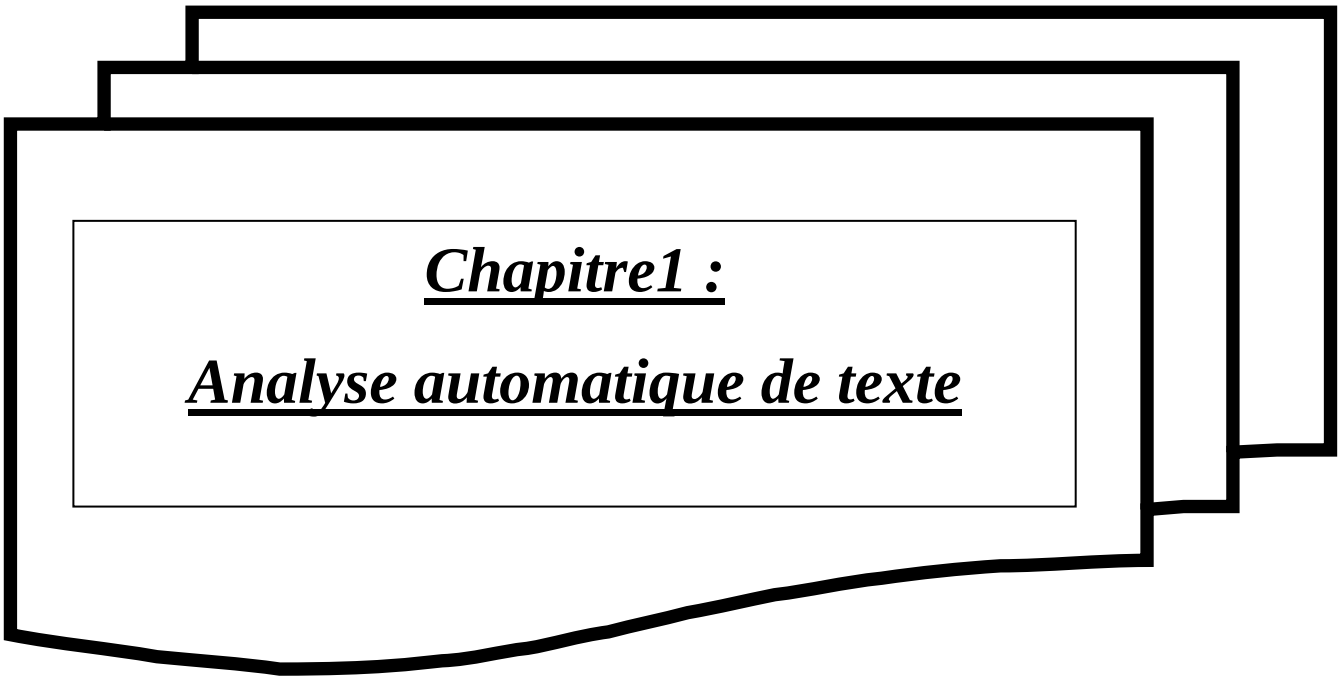
Dans ce chapitre, nous détaillons l'apprentissage en profondeur (définition, Pourquoi DL est-il utile ?, Les applications de l'apprentissage profond, comment ça marche …..), nous présentons la principale différence entre l'apprentissage en profondeur et l'apprentissage automatique **(*ML vs DL),*** Tout d'abord, nous présentons les réseaux de neurones et ces types, en fin nous détaillons le Deep Learning pour le texte et comment ça marche.

Chapitre III : Conception du système :

Dans cette partie du mémoire, nous décrivons la conception générale et détaillée de notre système. En détaillons chaque partie de celui-ci.

Chapitre IV : Implémentation :

Le dernier chapitre de ce mémoire est réservé aux résultats obtenus lors de l'implémentation du système que nous avons réalisé, ainsi on présente l'environnement (langages et outils) sur lequel le système sera validé et réalisé.



Chapitre1 :

Analyse automatique de texte

1.1 Introduction :

La grande masse de documents textuels disponibles actuellement ainsi que ses exploitations croissantes, nécessite le développement de méthodes et d'outils pour le traitement et l'exploration automatique du texte afin d'extraire le contenu le plus significatif.

Analyse de texte est l'une des techniques qui facilitent l'exploitation de la quantité volumineuse de documents textuels. Elle permet de construire une vision sur le contenu d'une ressource (morceau de texte, une page Web, une image, une séquence vidéo, etc.).

Analyse de documents textuels est une tâche très importante permet d'assigner à des entités qui se trouvant dans le texte des informations descriptives, explicatives, ou encore critiques.

1.2 Extraction de termes simples et composés :

L'extraction des unités significatives qui composent un texte, nécessite tout d'abord une récupération des éléments syntaxiques du document tels que les paragraphes, les sections, les phrases, les titres,... etc. Cette extraction consiste à analyser le contenu afin de détecter les informations les plus importantes.

Les méthodes d'extraction automatique de Termes visent à extraire automatiquement des concepts à partir d'un corpus. Ces méthodes sont essentielles pour l'acquisition des connaissances d'un domaine pour des tâches telles la recherche, la traduction, la classification,….. En effet, les concepts sont importants pour mieux comprendre le contenu d'un domaine. Ces termes peuvent être : composés d'un seul mot (généralement simple à extraire), ou composés de plusieurs mots (difficile à extraire).

1.2.1 Extraction de termes simples:

Un terme est une suite de caractères graphiques formant une unité sémantique et pouvant être distingués par un séparateur.

1.2.2 Extraction de termes composés :

Les mots composés sont souvent considérés comme des unités sémantiques et syntaxiques. Cette propriété rend donc indispensable leur recensement pour tenir compte du phénomène du figement dans le domaine du traitement automatique des langues.

Exemple :

Mark Elliot Zuckerberg (né le 14 mai 1984) est un entrepreneur et philanthrope américain sur Internet. Il est connu pour avoir co-fondé Facebook, Inc. et sert de son président, chef de la direction et actionnaire majoritaire. Il a également co-fondé et est membre du conseil d'administration du projet de développement de vaisseaux spatiaux à voile solaire Breakthrough Starshot. Zuckerberg a fait son entrée en bourse en mai 2012 avec une participation majoritaire. Sa valeur nette est estimée à près de 54 milliards de Dollars canadiens en mars 2020.

->Extraction de termes simples:

[Entrepreneur, philanthrope, américain, Internet, Facebook ……]

->extraction de termes composés :

Mark Elliot Zuckerberg -> nom de personne.

14 mai 1984 , mai 2012 ,mars 2020 ->Date.

Dollars canadiens ->->Monnaie d'État

->Dans le cas des termes simples extraction est facile, analyse en suite est très simple, mais dans le cas des termes composé analyse est très difficile.

1.3 Analyse syntaxique :

1.3.1 Définition *:*

D'une façon générale, l'analyse syntaxique sert à vérifier qu'une expression appartient au langage défini par une grammaire donnée. L'analyse syntaxique constitue un point clé dans un grand nombre de traitements automatiques, tels que la compréhension de texte, l'extraction d'information ou la traduction [1].

Le but d'un analyseur syntaxique est de pouvoir construire la structure syntaxique d'une phrase donnée en entrée. Autrement dit, sa tâche est d'associer, à la phrase découpée en unités, une représentation des groupements structurels entre ces unités ainsi que les relations de dépendance syntaxique des unités, telles que sujet-verbe ou verbe-objet.

C'est une tâche difficile, en raison de la complexité et de la richesse de la langue. [1][23]

1.3.2 Les étapes d'analyse syntaxique :

1.3.2.1 la Tokenisation :

La première étape de prétraitement indispensable est celle de tokenisation .Dans cette étape un texte est segmenté en unités élémentaires dites token. Un token peut être un mot ou une séquence de mots. Cette étape n'est pas complexe techniquement en français et en anglais où les mots sont le plus souvent séparés naturellement par la présence d'un espace ou d'une ponctuation. Cependant, le choix des règles de tokenisation impacte fortement les performances nécessite une expertise sur les données traitées **[2].**

Exemple :

Le text mining est un ensemble de techniques
[« Le » « text mining » « est » « un » « ensemble » « de » « techniques »]

1.3.2.2 Elimination des mots vides :

L'une des étapes importantes de l'extraction de mots significatifs est la suppression des mots vides. Les mots vides sont des mots peu significatifs et porteurs de peu de sens. De ce fait, l'élimination des mots vides a l'avantage évident de réduire le nombre de termes à extraire. Alors, le fait de ne pas éliminer les mots vides provoque inévitablement du bruit. On distingue deux techniques pour éliminer les mots vides :

->L'utilisation d'une liste de mots vides (aussi appelée anti-dictionnaire ou stoplist), la listes des mots vides contient les pronoms personnels, les prépositions, les articles... etc. exemple : He, she, help, and, is ...

-> L'élimination des mots dépassant un certain nombre d'occurrences dans la collection ou les mots rares de la collection **[22].**

Exemple :

[~~Le~~ *text mining* ~~est un~~ ensemble ~~de~~ techniques appartenant ~~au~~ domaine ~~de~~ ~~l~~'IA]

[*Text mining* ensemble techniques appartenant domaine IA]

1.3.2.3 La lemmatisation

Est l'analyse qui permet de regrouper les mots d'une même famille. Les mots d'une même famille détectés dans un texte sont donc réduits en une unique entité que l'on appelle un « lemme »ou « forme canonique d'un mot ». La lemmatisation consiste donc à regrouper

ensemble toutes les formes que peut prendre un mot unique. Ses formes peuvent être le pluriel, le verbe à l'infinitif, le verbe conjugué à tous les temps, le nom ... **[3].**

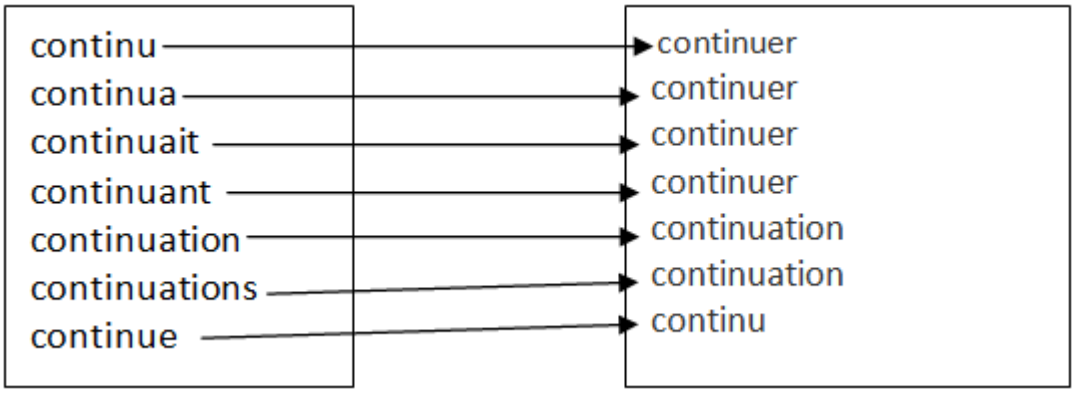


Figure 1.1 exemple de lemmatisation.

1.3.2.4 La Racinisation « Stemming » En Anglais :

Stemming est le processus de réduction des mots fléchis (ou parfois dérivés) à leur forme de racine, de base ou de racine de mot - généralement une forme de mot écrite. La tige n'a pas besoin d'être identique à la racine morphologique du mot; il suffit généralement que les mots associés correspondent à la même racine, même si cette racine n'est pas en soi une racine valide. Les algorithmes de stemming sont étudiés en informatique depuis les années 1960**[3]**

Exemple :

Prétraitement -> traite

économie, économiquement, économiste -> économ

Pour l'anglais : retrieve, retrieving, retrieval, retrieved, retrieves -> retriev

Impossible -> possible

Plays -> Play

Posting -> Post

L'inconvénient majeur de cette opération est qu'elle supprime dans certains cas la sémantique des termes originaux par exemple : traite, économ

1.3.2.5 Marquage (Tagging)

Une composante essentielle dans plusieurs domaines du **NLP** : résolution de coréférence, traduction automatique, recherche d'information, etc.

Cette technique cherche à localiser et classer les entités nommées dans un texte en catégories prédéfinies telles que : les noms de personnes, organisations, lieux, dates, quantités, pourcentages, heures, email ….etc.

Exemple :

Microsoft, Face book, Google -> organisation

Mark Elliot Zuckerberg -> personne

2008-06-29 -> Date

1.3.2.6 L'indexation automatique

L'indexation automatique est l'opération qui consiste à faire reconnaître par l'ordinateur des termes figurant dans le titre, le résumé, le texte complet (s'il est enregistré avec la notice documentaire) et parfois même l'indexation humaine, et à employer ces termes, soit tels quels soit après conversion en d'autres termes équivalents ou conceptuellement voisins, pour en faire des critères incorporés dans le fichier de recherche et utilisables pour retrouver le document. **[5]**

1.3.2.6.1 Les Types d'indexation automatique :

Il existe deux types d'indexation automatique ou semi-automatique :

-> le premier consiste à enrichir automatiquement l'indexation humaine par auto postage générique ou encore une indexation automatique non sélective (prise en compte de tous les mots non vides du document). Ce type d'indexation est utilisé de façon généralisée.

-> Le deuxième type d'indexation automatique est l'indexation automatique sélective, c'est-à-dire une prise en compte de certains termes seulement jugés par le système comme les plus représentatifs du contenu du document soit en langage naturel soit en langage contrôlé.

Une grande majorité des systèmes bâtis sur ce type d'indexation était encore en expérimentation jusqu'à l'intégration de module linguistique depuis une dizaine d'années environ. Pour mieux se rendre compte de l'importance de l'indexation automatique nous

allons tout d'abord voir la comparaison entre l'indexation humaine et l'indexation automatisée**. [7]**

Un index se compose:

-Un vocabulaire V, contenant tous les termes distincts de l'ensemble de documents, et pour chaque terme distinct, une liste inversée de publications.

-Chaque enregistrement stocke l'ID (désigné par id_j) du document d_j qui contient le terme et d'autres informations sur ce terme dans ce document.

-Pour chaque terme, nous avons une liste qui enregistre dans quels documents le terme se produit.

-Chaque terme de la liste est appelé classiquement **Posting (**publication).

-Un Posting est un tuple de la forme (t_i,d_j), où t_i est un identificateur de terme et d_j est un identifiant de document.

-Chaque Posting contient généralement:

- L'identifiant du document lié.
- La fréquence d'apparition du terme dans le document
- La position du terme pour chaque document (facultatif)

1.3.2.6.2 Calculer le poids :

Un document dans le modèle vectoriel est représenté par un vecteur de poids, dans lequel chaque poids de composant est calculé en fonction de certaines variations de:

A. TF

Dans le modèle TF, les coordonnées de vecteur du document d_j sont représentées par le nombre des occurrences d'un terme t^i, généralement normalisé avec le nombre des termes contenus dans ce document. **[8]**

Les différentes manières pour calculé TF :

$$TF(t_i,d_j) = \begin{cases} \frac{n_{ij}}{\sum_{k=1}^{k=m} n_{kj}} & si \quad n_{ij} > 0 \\ 0 \quad si \quad n_{ij} = 0 \end{cases}$$

$$TF(t_i,d_j) = \begin{cases} \frac{n_{ij}}{max(n_{kj})} & si \quad n_{ij} > 0 \\ 0 \quad si \quad n_{ij} = 0 \end{cases}$$

$$TF(t_i, d_j) = \begin{cases} \dfrac{n_{ij}}{1 + log\ (1 + log\ n_{ij})} & si \quad n_{ij} > 0 \\ 0 \quad si \quad n_{ij} = 0 \end{cases}$$

n(ij): nombres d'occurrences de terme(i) dans document(j).

max(nij) :max d'occurrences de terme(i) dans document(j).

ti : le terme i, dj : le document j.

B. IDF

Le schéma IDF consiste à réduire les coordonnées de certains axes, correspondant à des termes qui existent dans beaucoup documents.

Pour chaque terme t_i la mesure *IDF* est calculée en proportion des documents où t_i est apparu par rapport au nombre de documents du corpus.

$$idf_i = log \frac{N}{df_i}$$

N : est le nombre de tous les documents dans le corpus

dfi : est le nombre de documents dans lesquels le terme *ti* apparaît au moins une fois.

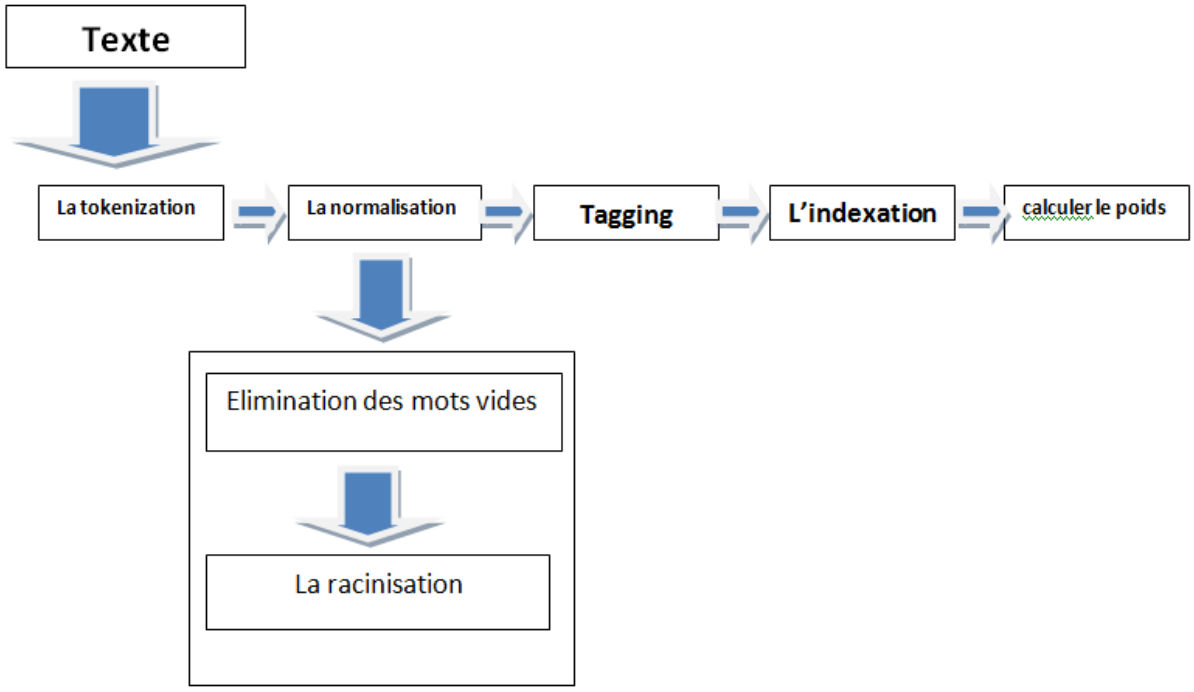


Figure 1.2 les étapes d'analyse syntaxique

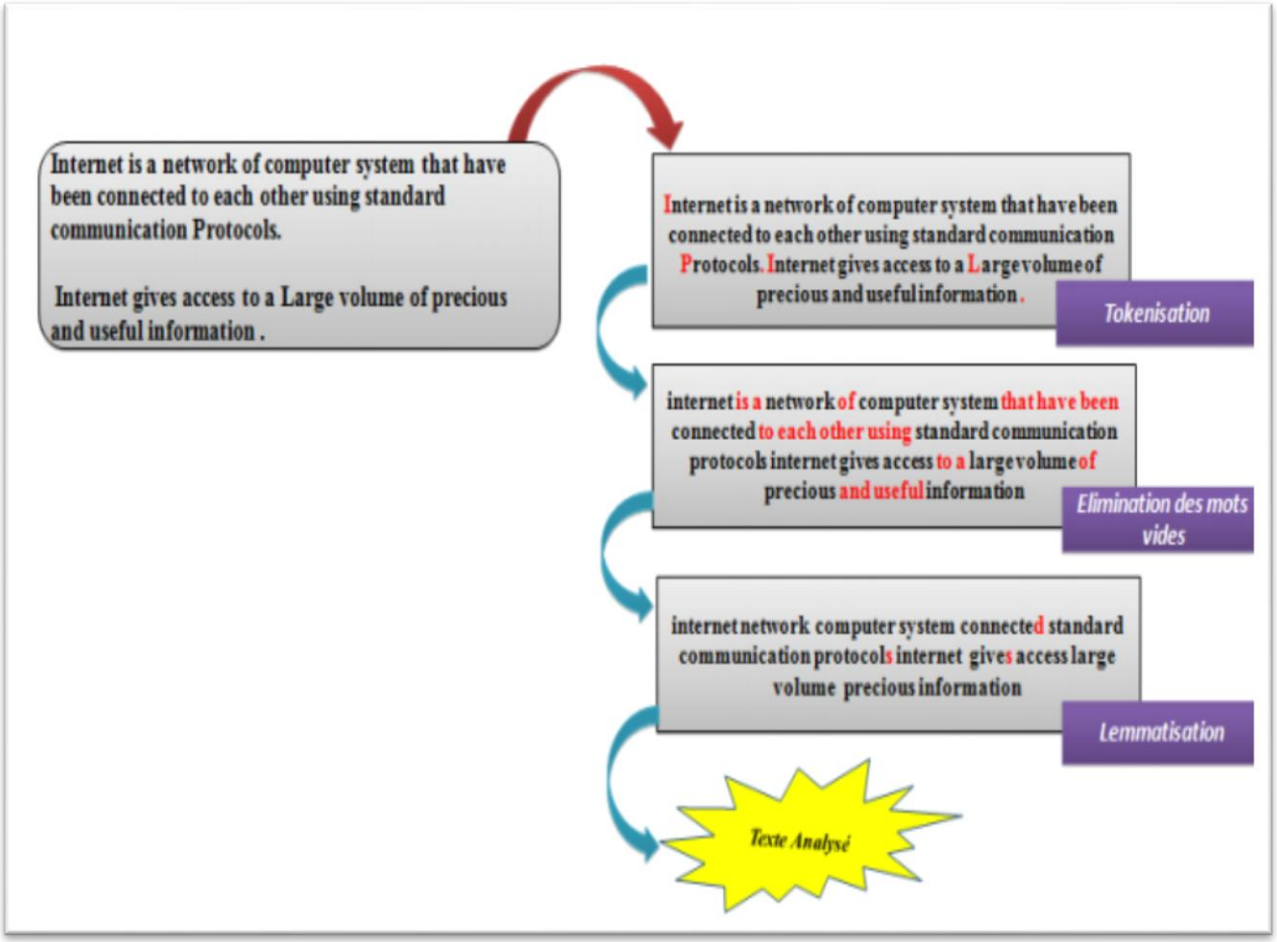


Figure 1.3 Exemple d'un texte analysé

1.4 Analyse sémantique:

1.4.1 Définition :

Il existe de nombreuses définitions qui diffèrent selon le domaine d'utilisation :

A. Définition 1

L'analyse sémantique du contenu en langue naturelle commence par la lecture de tous les mots du contenu pour saisir la véritable signification de tout texte. Il identifie les éléments de texte et les affecte à leur rôle logique et grammatical. Il analyse le contexte dans le texte environnant et il analyse la structure du texte pour lever sans ambiguïté le sens propre des mots qui ont plus d'une définition. **[9]**

B. Définition 2

L'analyse sémantique est par définition une technique qui consiste à se baser essentiellement sur le sens des phrases. Cette étude est en général effectuée afin de déterminer le sens exact des phrases. **[10]**

***C.* Définition 3**

La sémantique est définie comme une branche de la linguistique qui étudie les signifiés, ce dont on parle, ce que l'on veut énoncer. Sa branche symétrique, la syntaxe, concerne pour sa part le signifiant, sa forme, sa langue, sa graphie, sa grammaire… **[11]**

1.4.2 Pourquoi faire une analyse sémantique :

L'analyse sémantique s'avère extrêmement utile pour ceux qui souhaitent optimiser leur traitement de l'information non structurée. Lorsqu'il s'agit de collecter des informations ou des données, par exemple, la tâche peut s'avérer particulièrement complexe si ces dernières sont nombreuses. Grâce à l'utilisation de l'analyse sémantique, il est désormais possible d'automatiser le processus d'extraction des informations non structurée, afin d'augmenter considérablement le volume de traitement et la rapidité d'exécution.

Certaines entreprises par exemple, utilisent des solutions d'analyse sémantique pour trier des courriers envoyés au SAV et re-dispatcher les messages aux bons services.

1.4.3 Ressources sémantiques :

Une ressource sémantique est un vocabulaire contrôlé qui représente une liste de termes d'un domaine ou de plusieurs domaines qui ont été énumérés explicitement.

Les ressources sémantiques sont conçus afin d'organiser l'information et d'apporter une terminologie pour cataloguer et récupérer l'information. Les fonctions les plus importantes du vocabulaire contrôlé sont d'assembler une variante de termes et de synonymes à des concepts et de lier ensuite ces concepts soit dans un ordre logique ou selon un classement par catégories **[12].**

Ils existent différents types de ressources sémantiques telles que les taxonomies, les thésaurus, les ontologies, glossaires et dictionnaires.

1.4.3.1 Thésaurus :

Conçu dés la fin des années 1950, le thésaurus est un répertoire contenant une liste de mots fonctionnels, un dictionnaire des synonymes, des paraphrases, ainsi qu'une hiérarchie des termes (ou descripteurs). Ces descripteurs sont organisés de manière conceptuelle pour faciliter la description d'un domaine et harmoniser la communication et le traitement de l'information **[13].**

Les relations dans un thésaurus sont :

Synonymie : un terme X est le synonyme d'un terme Y.

Homonymie : un terme X a la même forme orale ou écrite qu'un terme Y alors qu'ils ont des sens différents.

Associative : un terme X est associé à un terme Y s'il y a une sorte où chaque terme appartient à une catégorie ou domaine. Ces relations peuvent être de nature très variée : (la cause et l'effet, le tout et sa partie, l'action et le lieu de l'action, l'objet et sa propriété, etc ...) **[14].**

1.4.3.2 Taxonomie :

Une taxonomie est une forme d'ontologie dont la grammaire n'a pas été formalisée. En d'autres termes les taxinomies et les thésaurus peuvent être considérés

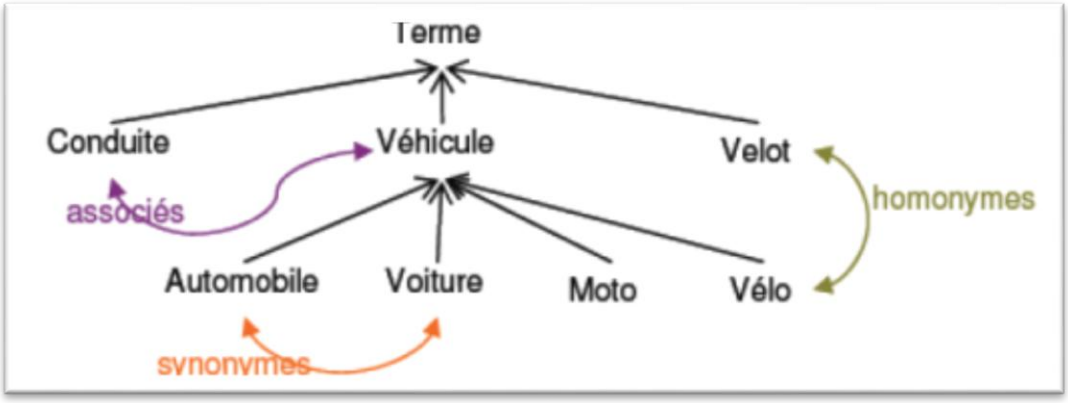


Figure 1.4 Les relations dans un thésaurus[14]

Comme des ontologies dont les relations seraient restées implicites car évidentes pour le lecteur (mais pas pour l'ordinateur). Une taxonomie est donc un type simple d'ontologie où toutes les relations sont de type "sorte-de", sans formalisation spécifique des propriétés **[15]**

1.4.3.3 Ontologie :

La notion d'ontologie intéresse à la fois l'ingénierie des connaissances, la linguistique et la philosophie. Ontologie, est un ancien terme qui été introduit en informatique depuis les années 1990, est un domaine de la philosophie concernant « l'étude de l'être en tant qu'être, c'est-à-dire l'étude des propriétés générales de ce qui existe ». Or, progressivement plusieurs définitions concernant cette notion sont apparues **[21].**

La connaissance dans les ontologies est principalement formalisée en utilisant les types de composants suivants : concepts (ou classes), relations (ou propriétés)/fonctions, axiomes (ou règles) et instances (ou individus) **[16].**

1.4.3.4 WordNet :

WordNet est une grande, base de données lexicale largement utilisée pour l'anglais. Ce lexique comporte environ 180000 termes organisés dans 117597 synsets qu'ils sont catégorisés en fonction de leurs catégories syntaxiques telles que le verbe, le nom, l'adjectif et l'adverbe. De plus, les mots et les sens des mots sont reliés les uns aux autres avec différents types de relations **[17].**

Cette ontologie a plusieurs définitions relatives aux besoins et aux objectifs utilisateurs, son origine remonte aux années 1986 à l'Université de Princeton où elle continue d'être développée et maintenue **[14].**

WordNet offre deux services distincts, Un vocabulaire décrivant les différents sens des mots. Une ontologie décrivant les relations sémantiques entre les mots **[14].**

Les synsets sont connectés en haut / bas de la hiérarchie par différents types de relations. Les types de relations les plus couramment utilisés pour mesurer la similarité sémantique sont : les « synonymes »sont les relation de base dans WordNet, les relations «hyperonymie» (indiquant qu'un synset est un type plus générale d'un autre synset), les relations « Est-un »pour les « hyponyme »(indiquant qu'un synset est un sous-type d'un autre synset), et les relations « Partie-de », pour les « holonyme »(indiquant qu'un synset est une partie d'un autre synset) et les relations « Antonyme »qui représente la relation entre un nom et son opposé**[17].**

Par exemple, « un chien est un type de mammifère »il est possible de dire que « un mammifère est un type d'animal ». Alors, l'animal est hyperonyme de mammifère, qui est aussi l'hyperonyme de chien.

1.4.4 Similarités entre concepts :

La similarité est au cœur de plusieurs travaux dans différents domaines tels que l'analyse de données, le raisonnement à partir de cas, la reconnaissance des formes, la résolution de problèmes, l'apprentissage, le transfert, ...etc. La similarité sémantique est une capacité abstraite qui exprime une liaison ou une relation entre deux concepts.

Il est évident pour un humain que les concepts « stylo »et « papier »sont liés beaucoup plus que « température »et « chaise ». Cet état de fait, difficile à formaliser sans recours aux ressources sémantiques : les ontologies permettent de montrer les liens entre les concepts (hyperonymie, antonymie, etc.) [14].

1.4.4.1 Techniques de calcul des mesures de similarité sémantique :

L'estimation de la similarité du texte sémantique est un problème de recherche qui vise à calculer les similarités entre les textes en fonction de leur signification et de leur contenu sémantique, plutôt que de leur représentation syntaxique **[17].**

La similarité sémantique est une évaluation du lien sémantique entre deux concepts dont le but est d'estimer le degré par lequel les concepts sont proches dans leur sens. La similarité entre deux concepts est liée aux caractéristiques qu'ils ont en commun (plus ils ont de caractéristiques communes, plus les concepts sont similaires). La similarité maximale est obtenue lorsque deux concepts sont identiques **[13] [17].**

Plusieurs mesures ont été définies pour le calcul de la similarité entre concepts.

Ces principales mesures sont classées par rapport aux caractéristiques des concepts permettant de calculer et d'évaluer la similarité. Ces caractéristiques se basent sur la structure à travers la longueur du chemin entre les deux nœuds de concept dans l'ontologie, ou sur l'information contenue par les concepts, ou sur les deux **[18].**

1.4.4.1.1 Mesures de similarité basées sur la structure d'ontologie :

Cette technique de calcul de similarité par la longueur du chemin ou le nombre d'arcs considère que la similarité entre deux concepts peut être calculée à partir du nombre de liens qui séparent les deux concepts. Ces mesures se servent de la structure hiérarchique de l'ontologie pour déterminer la similarité sémantique entre les concepts **[18].**

Parmi ces mesures de similarité sémantique on peut citer : la mesure de WuPalmer, la mesure de Leacock et Chorodow, la mesure de Rada et al, et la mesure du edge counting.

1.4.4.1.1.1 La mesure de Wu-Palmer :

La mesure de similarité WP (Wu-Palmer, 1994) observe la position de deux concepts C1 et C2 dans la hiérarchie de concepts par rapport à la position du concept commun C qui les subsume. La similarité de WP est mesurée comme deux fois la profondeur du concept commun le plus spécifique des concepts c1 et c2 sur la somme des profondeurs de c1 et c2 **[17].**

1.4.4.1.1.2 La mesure de Leacock et Chorodow :

La mesure de Leacock et Chodorow (Leacock and Chodorow, 1998) cette mesure basée sur le chemin et dépend de la longueur du plus court chemin entre concepts dans une hiérarchie **[18]**

1.4.4.1.2 Mesures de similarité basées sur le contenu en information des concepts :

La notion de contenu informationnel (IC) a été pour la première fois introduite par Resnik.

La fonction de profondeur du concept et la fréquence du concept dans un corpus donner une idée de la spécificité du concept. Avec la motivation de ces faits, IC est utilisé pour mesurer la similarité sémantique entre les concepts **[17].**

Parmi les mesures basées sur le contenu informationnel on peut citer : La mesure de Resnik et La mesure de Lin.

1.5 Les méthodes d'extraction des entités textuelles : [29]

L'extraction de caractéristiques de texte joue un rôle crucial dans la classification de texte, influençant directement la précision de la classification de texte. Il est basé sur VSM (modèle d'espace vectoriel, VSM), dans lequel un texte est considéré comme un point en N dimensions l'espace.

Le référentiel de chaque dimension du point représente une caractéristique du texte. Et le texte comporte généralement utilisez un ensemble de mots clés. Cela signifie que sur la base d'un groupe de mots-clés prédéfinis, nous calculons le poids des mots par certaines méthodes puis former un vecteur numérique. Il existe des méthodes d'extraction des caractéristiques comprennent la filtration, la fusion, la cartographie et la méthode de regroupement.

1.5.1 Indexation des concepts :

Dans la classification des textes, CI (indexation conceptuelle) [37] est une méthode de dimensionnalité simple mais efficace réduction. En prenant le centre de chaque classe comme sous-espace de structure vectorielle de base (sous-espace CI), et puis en mappant chaque vecteur de texte sur ce sous-espace, le la représentation des vecteurs de texte dans ce sous-espace est acquis.

Le montant du classement inclus dans ensembles de formation est exactement la dimensionnalité de CI sous-espace, qui est généralement plus petit que celui de la espace vectoriel de texte, donc réduction de la dimensionnalité de l'espace vectoriel est atteint. Chaque centre de classe généralisation des contextes de texte dans une classification peut être considéré comme un «concept», et la cartographie processus de vecteur de texte peut être considérée comme un processus d'indexation dans cet espace conceptuel.

1.6 Travaux connexes :

Plusieurs travaux menés dans le cadre de l’extraction de descripteurs sémantiques, parmi ces travaux nous citons ceux qui utilisent l’extraction pour la recherche d’information, le résumé automatique, l’annotation et la classification des documents... etc.

Dans ces travaux, l’extraction de descripteurs sémantiques nécessitent de disposer une ressource sémantique (ontologie, thésaurus...) afin d’extraire les concepts associés aux termes.

Les approches existantes de l'extraction des termes et des concepts sont basées sur les propriétés de la langue naturelle. De ce fait, elles sont dites approches linguistiques.

Dans cette section nous allons présenter quelques travaux concernant l'extraction de descripteurs.

1.6.1 Les travaux de L. Moncla et M. Gaio :

L'approche proposée par L. Moncla et M. Gaio (2017) **[20]**, permet d'annoter automatiquement des entités nommées (EN) et des informations spatiales associées, dans des textes descriptifs. Dans ce cas les annotations sont considérés comme des méta-donnés décrites un type d'élément prédéfini.

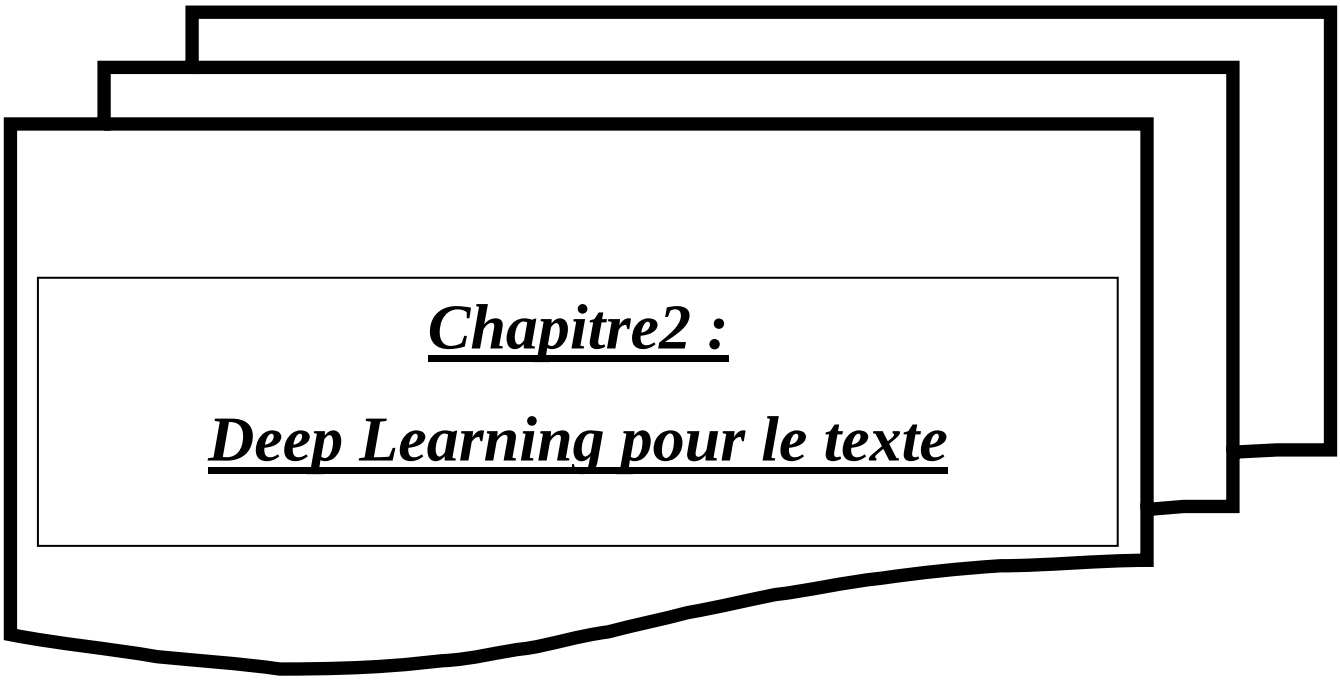
L. Moncla et M. Gaio (2017) **[20]**, présentent deux types d'approches pour l'annotation automatique des EN : les approches linguistiques ou symboliques et les approches probabilistes centrées sur les données et les techniques d'apprentissage. L'approche linguistique repose sur la description lexicale et syntaxique des syntagmes recherchés. Les EN sont repérées grâce à la construction de patrons lexico-syntaxiques utilisant des marqueurs lexicaux, et des dictionnaires.

1.7 Conclusion

Dans ce chapitre nous avons passé en revue analyse de texte et nous avons expliqué les étapes importantes pour faire une analyse complète de document non-structuré.

Nous avons expliqué comment les concepts composés sont extraits et préservés et comment faire une analyse syntaxique et sémantique pour essayer de comprendre le texte.

Dans le chapitre suivant, nous avons expliqué le Deep Learning et comment appliqué cette technique pour analyse de texte.



Chapitre2 :

Deep Learning pour le texte

2.1 Introduction :

Le traitement automatique de la langue naturel (TALN) est une tâche extrêmement difficile en informatique. Les langues présentent une grande variété de problèmes qui se changent d'une langue à l'autre.

Auparavant, les informaticiens divisaient une langue en ses formes grammaticales, telles que des parties de discours, des phrases, etc., utilisant des algorithmes complexes. Aujourd'hui, l'apprentissage en profondeur est la clé pour effectuer les mêmes taches.

2.2 Intelligence Artificielle :

L'intelligence Artificielle (IA) est la science dont le but est de faire faire par une machine des tâches que l'homme accomplit en utilisant son intelligence. La terminologie malheureuse d'Intelligence Artificielle est apparue en 1956. On peut lui préférer celle d'Informatique Heuristique. On ne parlera pas dans ce cours de machine intelligente, ni de programme intelligent. **[24]**

2.3 L'apprentissage Automatique (Machine Learning):

2.3.1 Définition :

L'apprentissage automatique est un sous-ensemble de l'intelligence artificielle (IA), qui est elle-même un sous-ensemble de la science des données. Il concerne les analyses descriptives, diagnostiques, prédictives et prescriptives. L'analyse descriptive se rapporte à ce qui s'est passé; l'analyse diagnostique explique pourquoi c'est arrivé; l'analyse prédictive permet de prévoir ce qui est le plus susceptible de se produire à l'avenir; et l'analyse prescriptive recommande le plan d'action le plus logique pour atteindre le résultat souhaité. L'apprentissage automatique est axé sur l'analyse prédictive et prescriptive, en fonction de la nature de l'analyse et des algorithmes utilisés. Cette section donne un aperçu des types les plus courants d'apprentissage automatique.

L'apprentissage automatique est une méthode d'entraînement d'algorithmes pour permettre à ces derniers d'apprendre à prendre des décisions et à faire des prévisions sans recevoir d'instructions découlant d'une programmation explicite. Il s'agit donc de fournir de très nombreuses données à un algorithme et de lui permettre d'en apprendre plus sur les informations traitées**. [27]**

Objectif : extraire et exploiter automatiquement l'information présente dans un jeu de données.

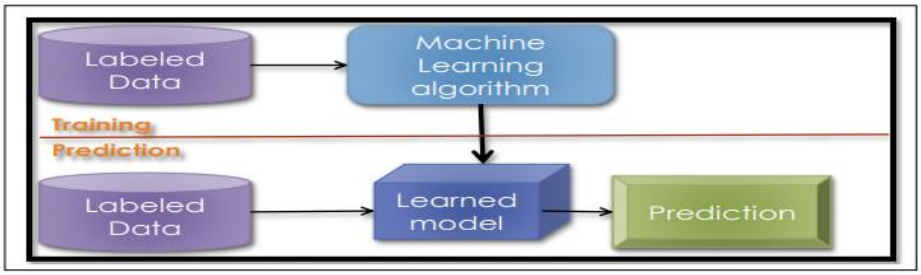


Figure 2.1 : Méthodes permettant d'apprendre et de prédire des données

2.3.2 Les Types D'apprentissage:

La méthode de paramétrage des apprentissages est une caractéristique importante pour distinguer différents types de réseaux de neurones.

2.3.2.1 L'apprentissage Supervisé :

Dans ce type d'apprentissage, les entrées et les sorties sont fournies au préalable. Ensuite, le réseau traite les entrées et compare ses résultats aux sorties souhaitées. Les poids sont ensuite ajustés grâce aux erreurs propagées à travers le système. Ce processus se produit à plusieurs reprises tant que les poids sont continuellement améliorés. L'ensemble de données qui permet l'apprentissage est appelé l'ensemble d'apprentissage. **[26]**

Exemple : classification des emails avec des emails déjà étiquetés.

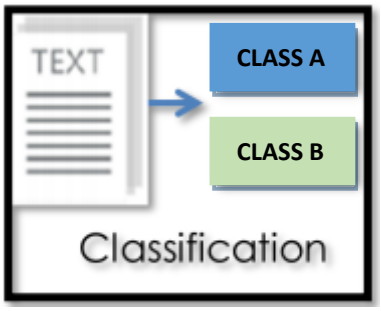


Figure 2.2 Exemple classification

2.3.2.2 L'apprentissage Non Supervisé :

Dans l'apprentissage non supervisé, le réseau est fourni avec des entrées mais pas avec les sorties souhaitées. Le système lui-même doit alors décider quelles fonctionnalités il utilisera pour regrouper les données d'entrée. C'est ce qu'on appelle souvent l'auto-organisation ou l'adaptation. **[26]**

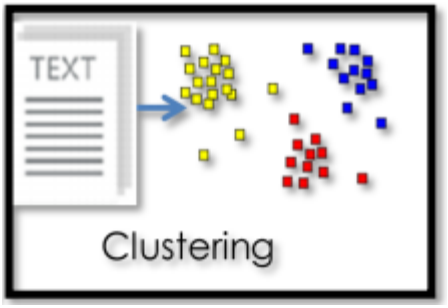


Figure 2.3 Exemple Clustering.

2.3.2.3 L'apprentissage Semi-Supervisé :

Dans l'apprentissage supervisé, l'étiquetage des données peut être long et coûteux. Si les étiquettes sont limitées, il est possible d'utiliser des exemples non étiquetés pour améliorer l'apprentissage supervisé. Étant donné que la machine n'est pas entièrement supervisée, on emploie le terme « semi-supervisé ». En ce qui concerne l'apprentissage semi-supervisé, on utilise des exemples non étiquetés et une petite quantité de données étiquetées pour améliorer la précision de l'apprentissage. **[27]**

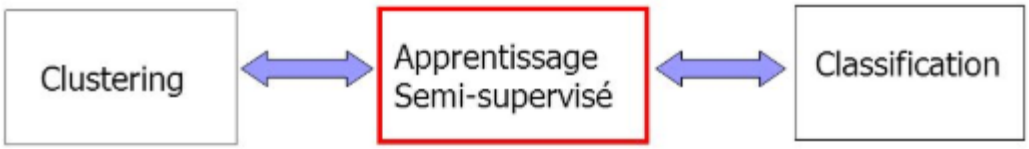


Figure 2.4 L'apprentissage Semi-supervisé

2.3.2.4 L'apprentissage Par Renforcement :

L'apprentissage par renforcement permet d'analyser et d'optimiser le comportement d'un agent en fonction du retour d'informations de l'environnement. Les machines essaient différentes situations pour déterminer les actions les plus avantageuses, plutôt que de simplement recevoir des instructions sur les actions à entreprendre. Ce qui distingue

l'apprentissage par renforcement des autres techniques, ce sont l'apprentissage par essais et erreurs et la récompense différée. **[27]**

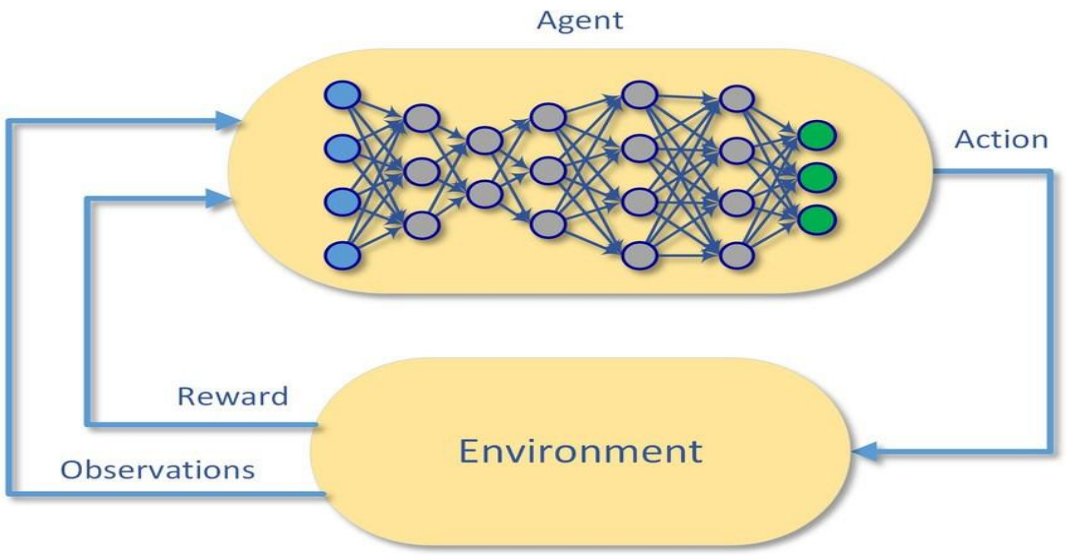


Figure 2.5 L'apprentissage Par renforcement

2.4 L`apprentissage Profond (Deep Learning) :

2.4.1 Définition :

L'apprentissage profond (deep learning) est un ensemble de techniques d'apprentissage automatique qui a permis des avancées importantes en intelligence artificielle dans les dernières années. Dans l'apprentissage automatique, un programme analyse un ensemble de données afin de tirer des règles qui permettront de tirer des conclusions sur de nouvelles données.

L'apprentissage profond est basé sur ce qui a été appelé, par analogie, des « réseaux de neurones artificiels », composés de milliers d'unités (les neurones) qui effectuent chacune de petites opérations simples. Les résultats d'une première couche de « neurones » servent d'entrée aux calculs d'une deuxième couche et ainsi de suite.

Par exemple : pour la reconnaissance visuelle, des premières couches d'unités identifient des lignes, des courbes, des angles… des couches supérieures identifient des formes, des combinaisons de formes, des objets, des contextes… Les progrès de l'apprentissage profond ont été possibles notamment grâce à l'augmentation de la puissance des ordinateurs et au développement de grandes bases de données (« big data »). **[26]**

2.4.2 Historique :

Les étapes majeures qui mènent à ce que nous avons maintenant. Ces étapes sont résumées dans le tableau suivant :

L'année	Contributeur	Contribution
300 AC	Aristotle	Introduction de l'associationnisme, début de l'histoire des humains qui essayent de comprendre le cerveau
1873	Alexander Bain	-Introduction du Neural Groupings comme les premiers modèles de réseaux de neurones
1943	McCulloch et Pitts	-Introduction du McCulloch Pitts (MCP) modèle considérer comme L'ancêtre des réseaux de neurones artificielles
1949	Donald Hebb	-Considérer comme le père des réseaux de neurones, il introduit la règle d'apprentissage de Hebb qui servira de fondation pour les réseaux de neurones modernes.
1958	Frank Rosenblatt	-Introduction du premier perceptron
1974	Paul Werbos	-Introduction de la retro propagation
1980	Teuvo Kohonen	-Introduction des cartes auto organisatrices
1980	Kunihiko Fukushima	-Introduction du Neocognitron, qui a inspiré les réseaux de neurone convolutif
1982	John Hopfield	-Introduction des réseaux de Hopfield
1985	Hilton et Sejnowski	-Introduction des machines de Boltzmann
1986	Paul Smolensky	-Introduction de Harmonium, qui sera connu plus tard comme machines de Boltzmann restreintes
1986	Michael I. Jordan	-Définition et introduction des réseaux de neurones récurrents
1990	Yann LeCun	-Introduction de LeNet et montra les capacités des réseaux de neurones profond

1997	Schuster et Paliwal	-Introduction des réseaux de neurones récurrents bidirectionnels
1997	Hochreiter et Schmidhuber	-Introduction de LSTM, qui a résolu le problème du vanishing gradient dans les réseaux de neurones récurrent
2006	Geoffrey Hinton	-Introduction des Deep Belief Network
2009	Salakhutdinov et Hinton	-Introduction des Deep Boltzmann Machines
2012	Geoffrey Hinton	-Introduction de AlexNet qui remporta le challenge ImageNet

Table 2-1: Le résumé de l'histoire de Deep Learning [28]

2.4.3 ML vs DL:

->La principale différence entre l'apprentissage en profondeur et l'apprentissage automatique est due à la façon dont les données sont présentées dans le système. Les algorithmes d'apprentissage automatique nécessitent presque toujours des données structurées, tandis que les réseaux d'apprentissage profond reposent sur des couches d'ANN (réseaux de neurones artificiels).

->Les algorithmes d'apprentissage automatique sont conçus pour «apprendre» à agir en comprenant les données étiquetées, puis à les utiliser pour produire de nouveaux résultats avec plus de jeux de données. Cependant, lorsque le résultat est incorrect, il est nécessaire de «les enseigner».

->Les réseaux d'apprentissage profond ne nécessitent pas d'intervention humaine, car les couches multi niveaux dans les réseaux de neurones placent les données dans une hiérarchie de différents concepts, qui finalement apprennent de leurs propres erreurs. Cependant, même ils peuvent se tromper si la qualité des données n'est pas assez bonne.

- > Les algorithmes d'apprentissage automatique nécessitent des données à puces, ils ne conviennent pas pour résoudre des requêtes complexes qui impliquent une énorme quantité de données.

->**Temps d'exécution :** Habituellement, un algorithme d'apprentissage profond prend beaucoup de temps à s'entraîner. En effet, il y a tellement de paramètres dans un algorithme d'apprentissage profond que leur formation prend plus de temps que d'habitude. L'algorithme d'apprentissage en profondeur de pointe ResNet prend environ deux semaines pour s'entraîner complètement à partir de zéro. Alors que l'apprentissage automatique prend relativement moins de temps pour s'entraîner, allant de quelques secondes à quelques heures.

Ce tour est complètement inversé au moment du test. Au moment du test, l'algorithme d'apprentissage en profondeur prend beaucoup moins de temps à s'exécuter. Alors que si vous le comparez avec k voisins les plus proches (un type d'algorithme d'apprentissage automatique), le temps de test augmente en augmentant la taille des données. Bien que cela ne soit pas applicable à tous les algorithmes d'apprentissage automatique, certains d'entre eux ont également de petits temps de test.

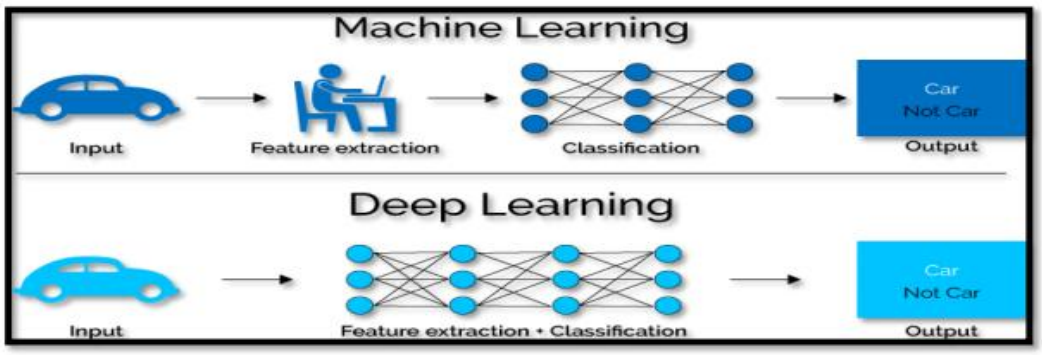


Figure 2.6 ML vs DL.

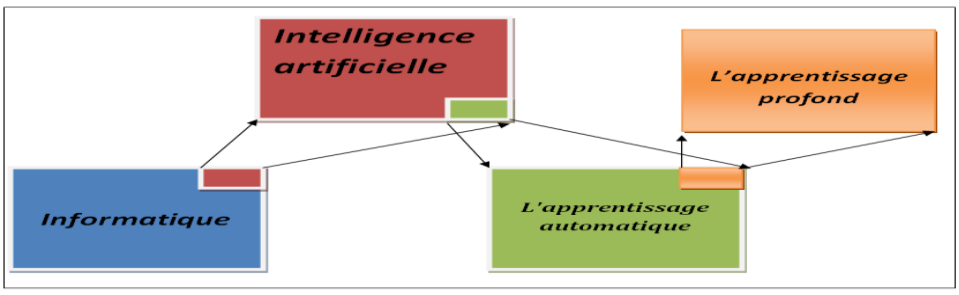


Figure 2.7 Schéma présentant la place de l'apprentissage automatique et du deep learning par rapport au domaine de l'informatique

2.4.5 Pourquoi DL Est-il Utile ?

1. Les fonctionnalités conçues manuellement sont souvent sur-spécifiées, incomplètes et prennent beaucoup de temps à concevoir et à valider.

2. Les fonctionnalités apprises sont faciles à adapter, rapides à apprendre.

3. L'apprentissage en profondeur fournit un cadre très souple, universel (presque) Et pouvant être appris pour représenter des informations mondiales, visuelles et linguistiques.

4. Peut apprendre à la fois sans surveillance et sous surveillance.

5. Apprentissage efficace du système conjoint de bout en bout.

6. Utiliser de grandes quantités de données d'entraînement.

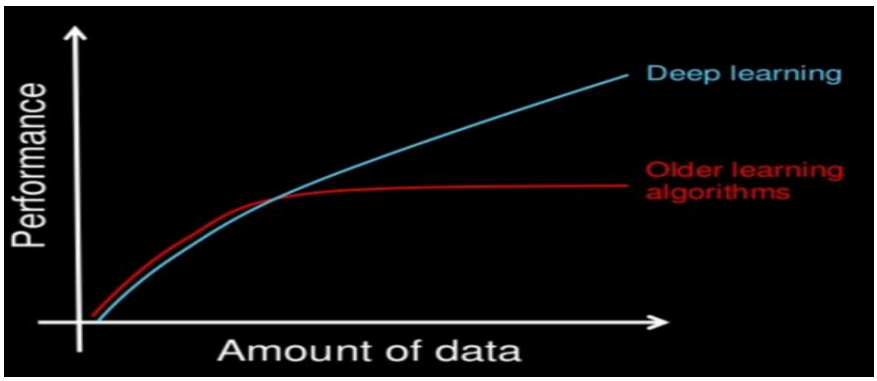


Figure 2.8 performance-amount of data

2.4.6 Les Applications De L'apprentissage Profond : [28]

a. Le Traitement Automatique De Langage Naturel : Le traitement automatique de langage naturel est une autre application du **DL**. Son but étant d'extraire le sens des mots, voire des phrases pour faire de l'analyse de sentiments. L'algorithme va par exemple comprendre ce qui est dit dans un avis Google, ou va communiquer avec des personnes via des chatbots. La lecture et l'analyse automatique de textes est aussi un des champs d'application du DL avec le Topic Modeling : tel texte aborde tel sujet.

b. Traduction Automatique : Il s'agit d'une tâche dans laquelle des mots, expressions ou phrases donnés dans une langue sont automatiquement traduits dans une autre langue. La traduction automatique existe depuis longtemps, mais DL permet d'obtenir les meilleurs résultats dans deux domaines spécifiques : -Traduction automatique de texte- Traduction automatique d'images La traduction de texte peut être effectuée sans aucun traitement

préalable de la séquence, ce qui permet à l'algorithme d'apprendre les dépendances entre les mots et leur correspondance avec une nouvelle langue.

c. Génération Automatique De Texte : C'est une tâche intéressante, où un corpus de texte est appris et à partir de ce modèle, un nouveau texte est généré, mot par mot ou caractère par caractère. Le modèle est capable d'apprendre comment épeler, ponctuer, former des phrases et même capturer le style du texte dans le corpus. Les grands réseaux de neurones récurrents sont utilisés pour apprendre la relation entre les éléments dans les séquences de chaînes d'entrée, puis pour générer du texte.

d. Analyse Des Sentiments Du Texte : De nombreuses applications ont des commentaires ou des systèmes de révision basés sur des commentaires intégrés à leurs applications. La recherche sur le traitement du langage naturel et les réseaux de neurones récurrents ont parcouru un long chemin et il est maintenant tout à fait possible de déployer ces modèles sur le texte de votre application pour extraire des informations de niveau supérieur. Cela peut être très utile pour évaluer la polarité sentimentale dans les sections de commentaires ou pour extraire des sujets significatifs à l'aide de modèles de reconnaissance d'entités nommées.

2.4.7 Le *Deep Learning*, Comment Ça Marche ?

Le *Deep Learning*, ou apprentissage profond, appartient à la grande famille de l'intelligence artificielle. Plus précisément, il constitue un sous-ensemble de la machine Learning et fait appel à des types particuliers de réseaux de neurones artificiels. Il présente donc des caractéristiques similaires à ces techniques, notamment la capacité d'apprentissage de façon autonome.

Mais la différence majeure de l'apprentissage profond réside dans la structure de ses neurones, disposés en couches. À l'intérieur de chacune d'entre elles, les neurones ne sont pas interconnectés. En revanche, ils sont tous reliés à ceux des couches précédentes et suivantes. La première couche reçoit les données en entrée, et la dernière fournit le résultat en sortie. Entre les deux, les couches intermédiaires sont dites « cachées ». Dans le schéma ci-dessus, on en compte une seule (en bleu). Et cette architecture confère des facultés spéciales. En effet, chaque couche permet une analyse de plus en plus approfondie des données d'entrée. Ainsi, le réseau établit lui-même une représentation de ce qu'il reçoit, qu'il s'agisse d'une image, d'un

texte, etc. Par exemple, à partir de portraits humains, le programme va d'abord distinguer le visage des cheveux, puis reconnaître le nez, la bouche, les yeux

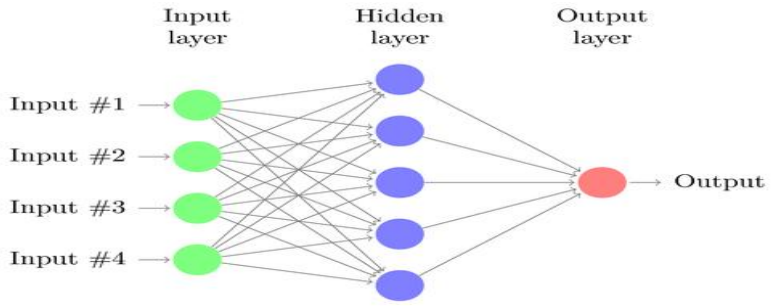


Figure 2.9 Le *deep Learning*, comment ça marche ?

2.4.8 Avantages De L'apprentissage Profond :

Contrairement aux algorithmes classiques, qui apprennent des solutions ne fonctionnant que pour un seul problème, les solutions trouvées par les réseaux de neurones profonds peuvent souvent être appliquées à d'autres tâches similaires (moyennant quelques adaptations). On appelle cela le Transfert Learning(en anglais). Lorsque l'on fait du Transfert Learning, on coupe en quelque sorte la partie du réseau qui est spécifique à une tâche particulière et on garde la partie généraliste. Un des avantages majeurs des réseaux de Deep Learning réside dans leur capacité à continuer à s'améliorer en même temps que le volume de vos données augmente**[30].**

2.4.9 Limites De L'apprentissage Profond :

Cela quelques limites de l'apprentissage profond :

-> Pour résoudre des problèmes de plus en plus complexes, il n'est pas suffisant d'ajouter toujours plus de couches. Les deux grosses problématiques de Deep learning;

-> La difficulté d'apprentissage;

-> La complexité calculatoire croissante avec le nombre de couches.

-> le Deep Learning exige une puissance de calcul considérable.

-> Entraînement coûteux en ressources (calcul, mémoire, ...).

-> L'établissement de réseaux de neurones profonds pose un défi: déterminer le nombre de couches cachées ainsi que le nombre de neurones par couches.

2.5 Les Réseaux De Neurones :

Les réseaux de neurones proposent une simulation du fonctionnement de la cellule nerveuse à l'aide d'un automate : le neurone formel. Les réseaux neuronaux sont constitués d'un ensemble de neurones (nœuds) connectés entre eux par des liens qui permettent de propager les signaux de neurone à neurone. Grâce à leur capacité d'apprentissage, les réseaux neuronaux permettent de découvrir des relations complexes non-linéaires entre un grand nombre de variables, sans intervention externe. De ce fait, ils sont largement utilisés dans de nombreux problèmes de classification (ciblage marketing, reconnaissance de formes, traitement de signal,…) d'estimation (modélisation de phénomènes complexes,…) et prévision (bourse, ventes,…). Il existe un compromis entre clarté du modèle et pouvoir prédictif. Plus un modèle est simple, plus il sera facile à comprendre, mais moins il sera capable de prendre en compte des dépendances trop variées**. [26]**

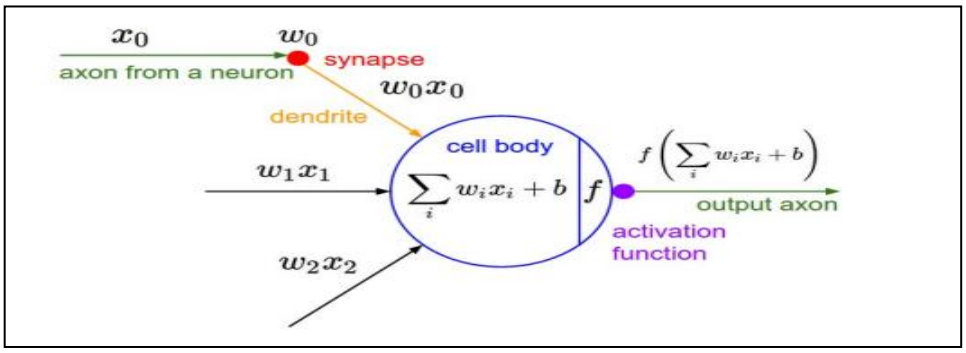


Figure 2.10
Un neurone Artificiel

Les **Xi** sont des valeurs numériques qui représentent soit les données d'entrée, soit les valeurs sorties d'autres neurones. Les poids Wi sont des valeurs numériques qui représentent soit la valeur de puissance des entrées, soit la valeur de puissance des connexions entre les neurones. Il existe des opérations qui se passent au niveau du neurone artificiel. Le neurone artificiel fera un produit entre le poids (w) et la valeur d'entrée (x), puis ajouter un biais (b), le résultat est transmis à une fonction d'activation (f) qui ajoutera une certaine non-linéarité.

2.6 Deep Learning Pour Le Texte:

Les premiers succès de l'utilisation de l'apprentissage en profondeur pour résoudre les problèmes de reconnaissance d'image ont conduit à des efforts visant à utiliser l'apprentissage en profondeur pour des fonctionnalités d'apprentissage dans d'autres domaines. Dans cette section, nous explorerons comment l'apprentissage profond a été appliqué pour résoudre des problèmes dans le texte. L'exploration ou le traitement de texte est un sous-ensemble du

traitement du langage naturel (NLP), tandis que l'apprentissage en profondeur peut résoudre efficacement l'analyse syntaxique et l'interprétation sémantique du texte. L'application de l'apprentissage en profondeur pour résoudre des problèmes liés au texte tente de résoudre le problème fondamental de la NLP, qui est la complexité inhérente à la représentation, à l'apprentissage et à l'utilisation des connaissances linguistiques qui est souvent influencée par les contextes et situations du monde réel. L'apprentissage profond a été utilisé pour représenter la morphologie d'un texte, ou en d'autres termes la représentation des parties constituantes d'un mot. Sans considérer la morphologie, les mots rares et complexes sont mal représentés en tant que représentation vectorielle. On utilisé un réseau neuronal récurrent (RNN) pour traiter le morphème des mots comme un vecteur, ce qui a permis d'améliorer les résultats par rapport aux techniques d'apprentissage non profond en termes d'estimation de mots rares et complexes. **[34]**

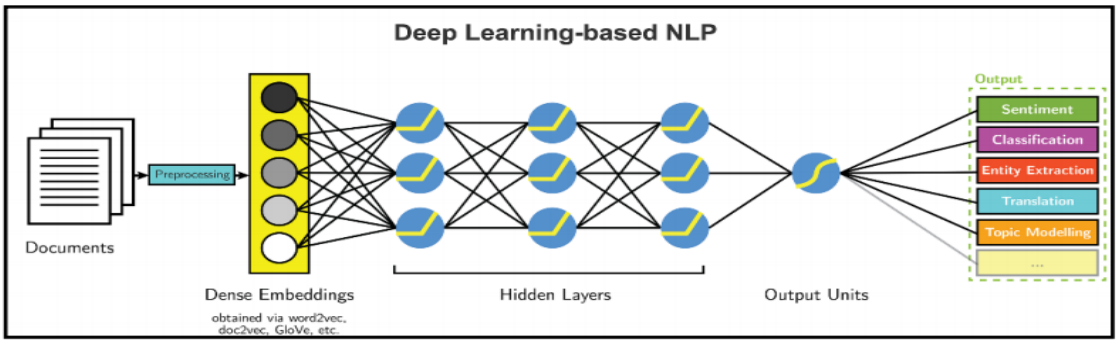


Figure 2.11 Exemple d'application du NLP avec le Deep Learning.

2.6.1 Les Algorithmes De L'apprentissage Profond Avec Le NLP :

La figure 2.12 ci -dessous représente des quelques algorithmes et leur utilisation avec le NLP

Algorithme	Utilisation du NLP
Neural Network - NN (feed)	▪ Part-of-speech Tagging ▪ Tokenization ▪ Named Entity Recognition ▪ Intent Extraction
Recurrent Neural Networks -(RNN)	▪ Machine Translation ▪ Question Answering System ▪ Image Captioning
Recursive Neural Networks	▪ Parsing sentences ▪ Sentiment Analysis ▪ Paraphrase detection ▪ Relation Classification ▪ Object detection
Convolutional Neural Network - (CNN)	▪ Sentence/ Text classification ▪ Relation extraction and classification ▪ Spam detection ▪ Categorization of search queries ▪ Semantic relation extraction

Figure 2.12 Algorithmes d'apprentissage et leur utilisation avec le NLP

2.6.2 Tâches Effectuées Par Deep Learning Pour L'analyse De Texte:

Dans cette section, nous présentons brièvement les quatre tâches effectuées par Deep Learning pour l'analyse de texte:

A. Part-Of-Speech Tagging :

Est une partie cruciale du traitement du langage naturel. Elle consiste à étiqueter les mots avec une partie du discours, a pour objectif de classer chaque mot avec un signe unique indiquant son rôle syntaxique (à associer à chaque mot d'un texte sa classe morphosyntaxique), par exemple: nom, un verbe, un adjectif,…etc. Le POS constitue la base de la résolution d'entité nommée, de l'analyse des sentiments, de la réponse aux questions, et l'ambiguïté du sens des mots.

Exemple:

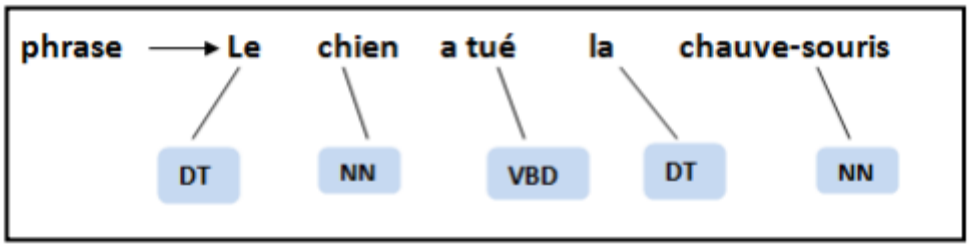


Figure 2.13 Exemple de Part-Of-Speech Tagging.

Liste du Tag & description:

La figure 2.16 ci-dessous représente des quelques Tags et leurs descriptions.

Tag	Description
CC	Coordinating conjunction
CD	Cardinal number
DT	Determiner
EX	Existential there
FW	Foreign word
IN	Preposition or subordinating conjunction
JJ	Adjective
JJR	Adjective, comparative
JJS	Adjective, superlative
LS	List item marker
MD	Modal
NN	Noun, singular or mass
NNS	Noun, plural
NNP	Proper noun, singular
NNPS	Proper noun, plural
PDT	Predeterminer
POS	Possessive ending
PRP	Personal pronoun

Figure 2.14 Tags et leurs descriptions

B. Named Entity Recognition (NER) :

Les étiqueteurs NER (ou reconnaissance d'entités nommées en français) marque les éléments atomiques de la phrase en catégories plus grandes telles que (noms de personnes, noms d'organisations ou d'entreprises, noms de lieux, quantités, distances, valeurs, dates, etc.).

Exemple : Exemple de Named Entity Recognition (la figure 2.15).

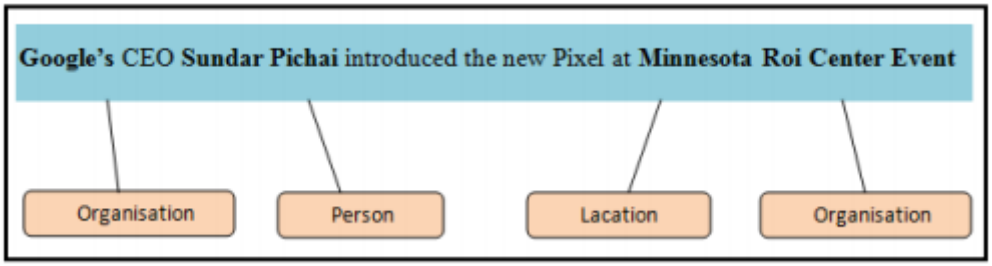


Figure 2.15 Exemple de la tache NER.

C. Semantic Role Labeling (SRL)

SRL vise à donner un rôle sémantique à un constituant syntaxique d'une phrase. Parsing ou Chunking est le processus qui consiste à déterminer la structure syntaxique d'un texte en analysant ses mots constitutifs sur la base d'une grammaire sous-jacente (du langage).

L'exemple de grammaire ci-dessous, où chaque ligne indique une règle de la grammaire à appliquer à un exemple de phrase «Tom a mangé une pomme».

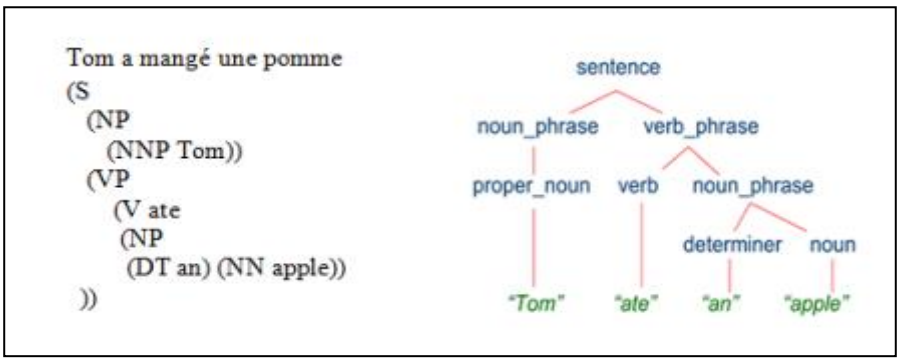


Figure 2.16 Exemple de la tache Parsing

D. Extraction des concepts composés:

Les approches d'apprentissage en profondeur sont des approches d'apprentissage supervisé qui sont construites par réseaux de neurones à plusieurs couches. Dans ce type d'approche, les réseaux de neurones récurrents (RNN) a été largement utilisé pour les tâches NLP dans divers domaines. Plusieurs études récentes ont utilisé RNN pour extraire des mots composés de textes avec différents degrés de complexité. Un RNN profond l'architecture a été proposée en ajoutant une couche commune pour déterminer les mots composés à partir de texte.

Les chercheurs ont utilisé la sortie de la première couche pour sélectionner les mots composés (Vrai ou Faux), puis appliqué le deuxième calque pour identifier la position des phrases clés (Unique, Begin, Middle, End and Not) Une étude de Meng et al. (2017) ont appliqué un modèle de codeur-décodeur RNN pour générer des phrases clés. **[35]**

2.6.3 Les Phases D'analyse De Texte En Utilisant Deep Learning :

2.6.3.1 Phase 1: Collection de données :

Il s'agit de créer un modèle à base d'un corpus de documents, ce corpus est partitionné en deux ensembles : l'ensemble d'apprentissage et celui de test. Cela consiste, en premier temps, à entrainer ce modèle avec l'ensemble d'apprentissage, et une fois appris, nous testerons son efficacité avec l'ensemble test. Dans certains cas, nous terminerons ce processus par une étape de validation du modèle avec un ensemble de nouveaux documents.

2.6.3.2 Phase 2 Préparation de données :

A. Phase 2.a Prétraitement

Chaque document de la collection de données va passer par les 3 étapes (Tokenisation, normalisation, la suppression de Stopword) et on va obtenir des résultats comme illustrait la figure 2.18 .

B. Phase 2.b L'extraction des termes importants et les termes composés:

A partir des documents textuels nettoyés et normalisés, on va obtenir les termes (simples et composés) extraits de la collection de données **.**

Les termes-clés sont les mots ou les expressions poly-lexicales qui représentent le contenu principal d'un document. Ils sont utiles pour diverses applications, telles que l'indexation automatique ou le résumé automatique, mais ne sont pas toujours disponibles. De ce fait, nous nous intéressons à l'extraction automatique de termes composés et, plus particulièrement, à la difficulté de cette tâche lors du traitement de documents appartenant à certaines disciplines scientifiques.

2.6.3.3 Phase 3: Modèle d'apprentissage :

La dernière étape du cadre d'apprentissage consiste à former un modèle à l'aide des fonctionnalités créées à l'étape précédente (étape de la préparation des données).

Le modèle d'apprentissage profond contient principalement trois types de couches: la couche d'entrée, Plusieurs couches cachées, la couche de sortie.

A. Phase 3.a: Transformation de données :

Comme tous les modèles d'apprentissage nécessitant une transformation de données dans la couche d'entrée ,Comme nous le savons déjà, les machines ou les algorithmes ne peuvent pas comprendre les caractères / mots ou les phrases, ils ne peuvent prendre que des nombres en entrée qui incluent également des fichiers binaires. Mais la nature inhérente des données textuelles est non structurée et bruyante, ce qui rend impossible toute interaction avec les machines. Les performances et la précision des algorithmes d'apprentissage automatique et d'apprentissage approfondi dépendent fondamentalement du type de technique d'ingénierie de caractéristiques utilisée. **[36]**

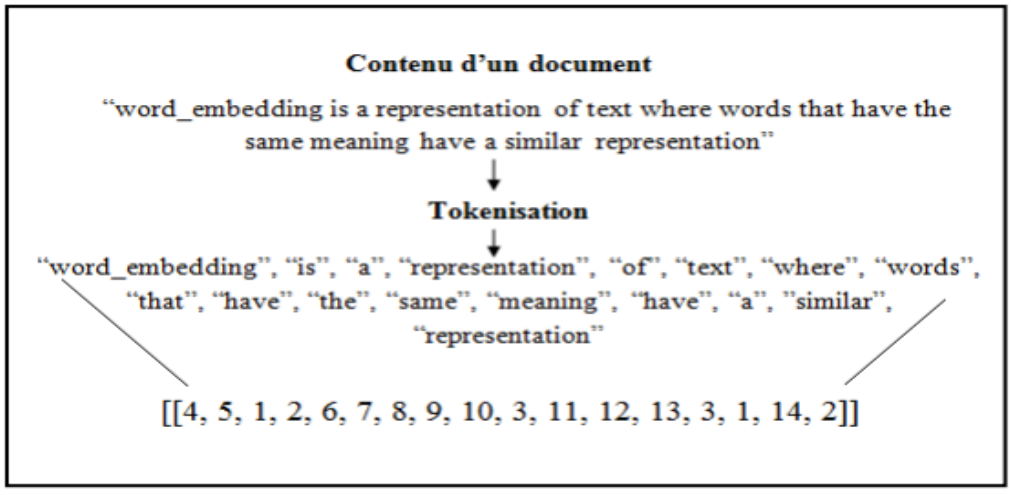


Figure 2.17 : illustration de transformation de données.

B. Phase 3.b: Codage de données :

Dans chaque tâche d'analyse de texte, la première et la plus importante étape est de parvenir à une représentation efficace du texte.

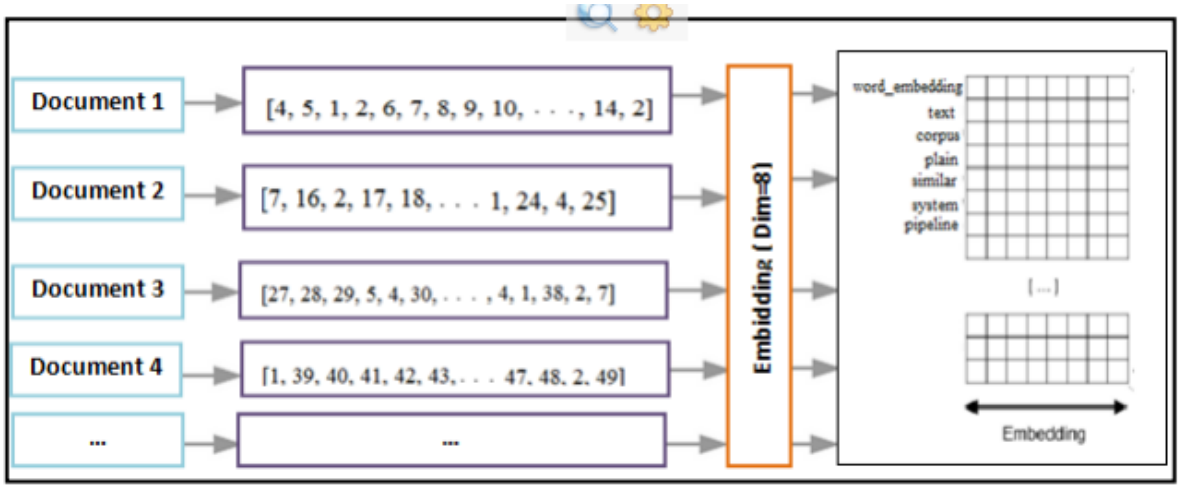


Figure 2.18 représentations de texte.

C. Phase 3.c Choix l'algorithme :

Il existe de nombreux choix de modèles d'apprentissage profond qui peuvent être utilisés pour former un modèle final. Nous allons choisir l'algorithme LSTM (Long Short - Term Memory en anglais).

2.7 Conclusion :

L'apprentissage profond est le domaine le plus émergeant de l'apprentissage automatique et a apporté une contribution importante dans divers domaines de recherche. Cela a permis de surmonter les inconvénients des méthodes traditionnelles en rendant les systèmes moins complexes et plus rapides. L'apprentissage profond a été utilisé avec analyse de texte dans plusieurs domaines de recherche, ce qui est très prometteur et constitue un succès. Dans ce chapitre nous avons exposé le technique de l'apprentissage profond, ainsi que ses avantages, et ses limites, le traitement du langage naturel, enfin les techniques de l'apprentissage profond avec le texte.

Chapitre3 :

Conception du système

3.1 Introduction :

La conception est le processus qui consiste à représenter les diverses fonctions du système, c'est certainement la partie la plus importante de notre travail. Dans ce chapitre nous allons présenter une architecture globale et une architecture détaillée où nous allons expliquer en détail le fonctionnement du notre système.

3.2 Architecture Générale :

La figure 3.1 présente le principe général de fonctionnement de notre système. Le but du système qui nous allons réaliser est de repérer les concepts les plus représentatifs à partir des documents pour pouvoir générer des nouvelles informations enrichir le contenu textuel.

Ce schéma montre que le système se décompose en trois modules principaux :

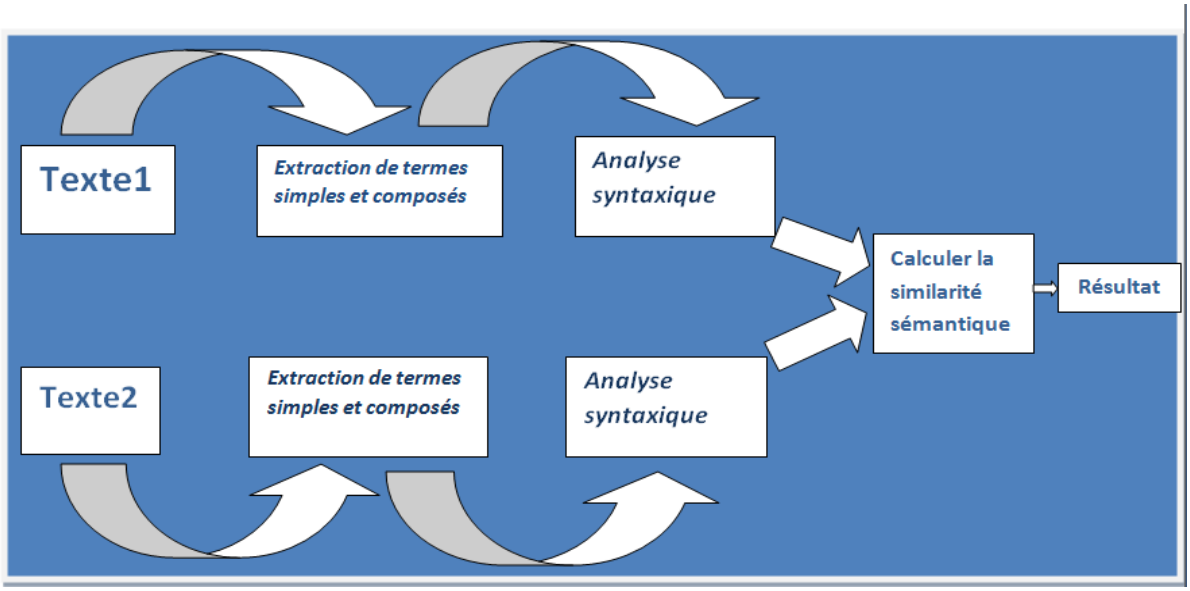


Figure 3.1 Architecture globale du système

3.2.1 Le module d'extraction de termes simples et composés :

L'extraction des unités significatives qui composent un texte, nécessite tout d'abord une récupération des éléments syntaxiques du document tels que les paragraphes, les sections, les phrases, les titres,... etc. Cette extraction consiste à analyser le contenu afin de détecter les informations les plus importantes.

3.2.2 Le module d'analyse syntaxique :

Cette étape consiste à analyser le contenu du document, nettoyer le texte à partir de mots non significatifs, et cherche à localiser et classer les entités nommées dans un texte en catégories prédéfinies telles que : les noms de personnes, organisations, lieux, dates, quantités, pourcentages, heures, email ….etc.

3.2.3 Le module calculer la similarité sémantique :

La similarité entre deux concepts est liée aux caractéristiques qu'ils ont en commun (plus ils ont de caractéristiques communes, plus les concepts sont similaires). La similarité maximale est obtenue lorsque deux concepts sont identiques.

3.3 Conception Détaillée :

Dans cette section nous allons essayer de détailler les différents modules composant ce système. La figure (3.2) présente l'architecture détaillée du système.

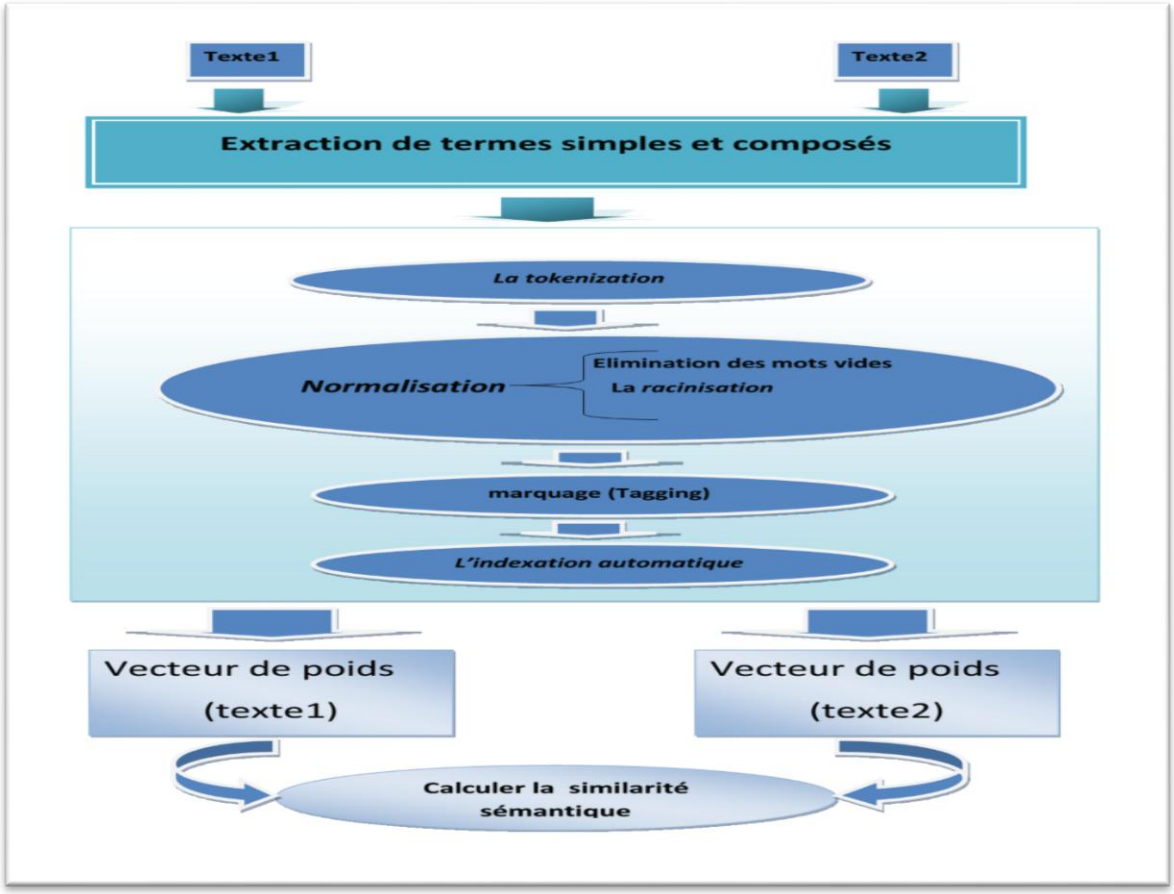


Figure 3.2 Architecture détaillée du système

3.3.1 Présentation de la collection :

Dans nos expérimentations, notre système consiste à extraire les meilleurs concepts des termes composés et faire analyse syntaxique (la tokenization, la normalisation, marquage, l'indexation automatique) en fin calculer la similarité sémantique entre deux textes.

3.3.2 Extraction des termes simples et composés :

Nous sommes intéressés dans la première phase de notre système consiste à extraire les termes simples et composés pouvant servir à représenter le contenu des documents, et ceci est dicté par le fait que les mots composés sont précis et moins ambigus que les mots simples.

Les termes composés sont construits en combinant deux ou plusieurs termes simples, par exemple, « triple »est un terme d'un seul mot, mais "triple pontage coronarien" est un terme composé de trois mots. Il est important de reconnaître les mots composés, car ce sont des unités de sens, par exemple : « Arbre à cames »ou « pomme de terre », extraites comme expressions, et non mot par mot. L'objectif consiste à définir une méthode d'acquisition de termes complexes à partir de corpus pouvant servir à représenter au mieux le contenu des documents. Le plus souvent un ensemble de patrons syntaxique comme [nom + nom], [nom + prep +nom] ou [adjectif + nom] est utilisé pour l'identification. Alors, nous partons de l'idée de la nature des mots composés, on a besoin d'une analyse grammaticale des mots (verbes - noms – pronoms- adjectifs, etc.) dans cette phrase.

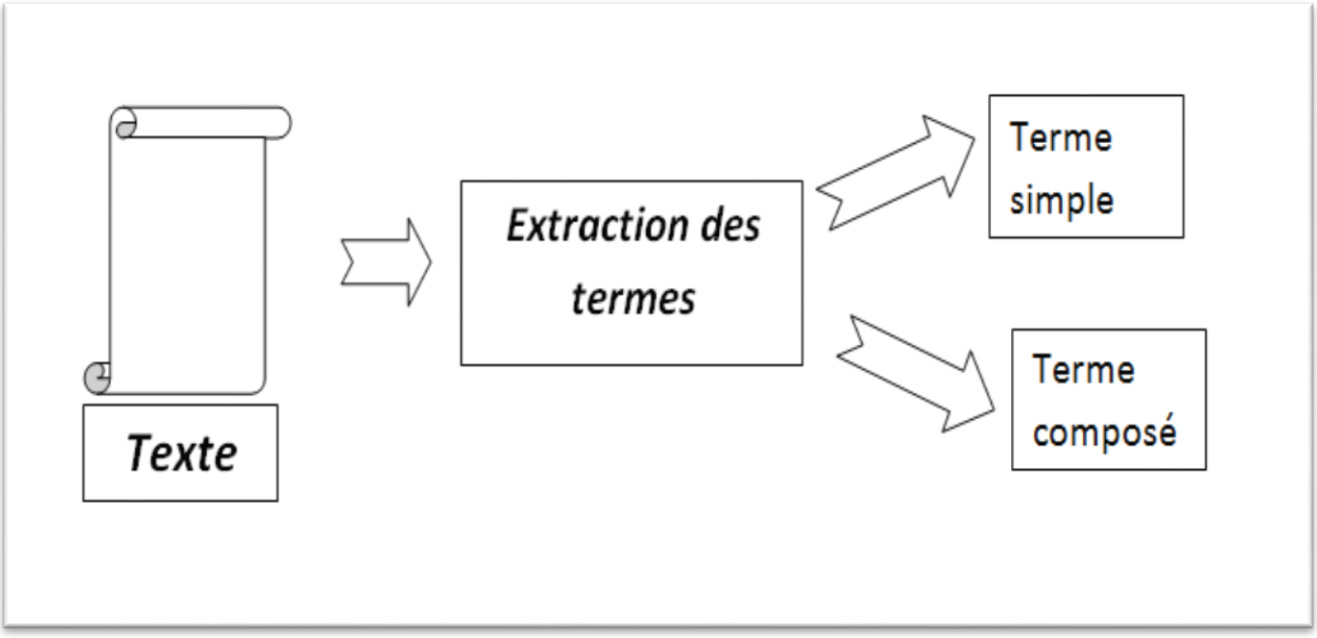


Figure 3.3 Extraction des termes à partir un texte.

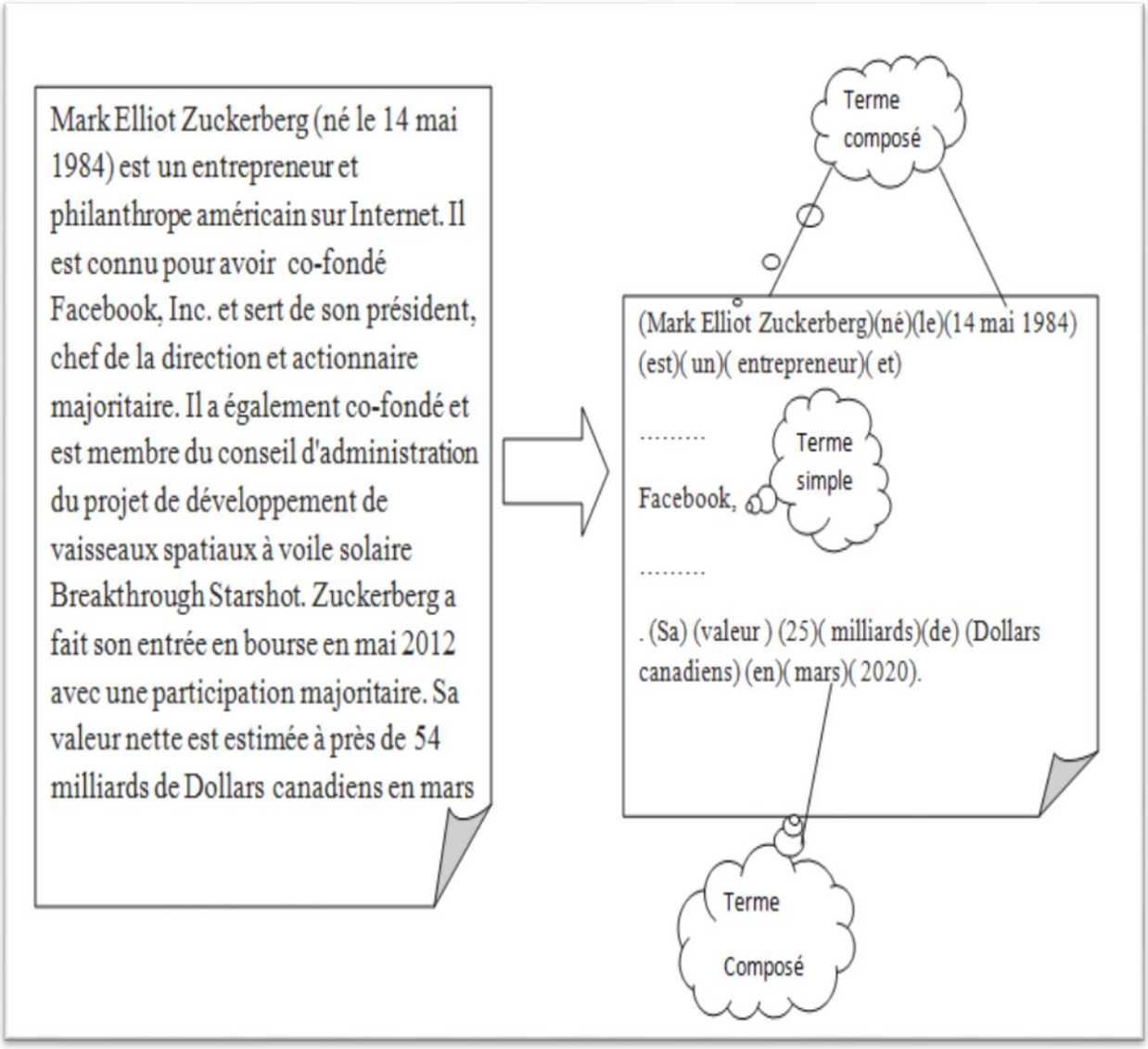


Figure 3.4 exemple d'extraction des termes simple et composé

3.3.3 Analyse syntaxique :

Pour effectuer une bonne extraction des différentes composantes des documents dans le système, cela consiste à analyser chaque document de la collection afin d'extraire un ensemble de mots-clés qui représentent aux mieux le contenu de ces documents. Le processus de l'analyse se compose de plusieurs étapes que nous avons schématisées dans la figure (3.5) ci-dessous.

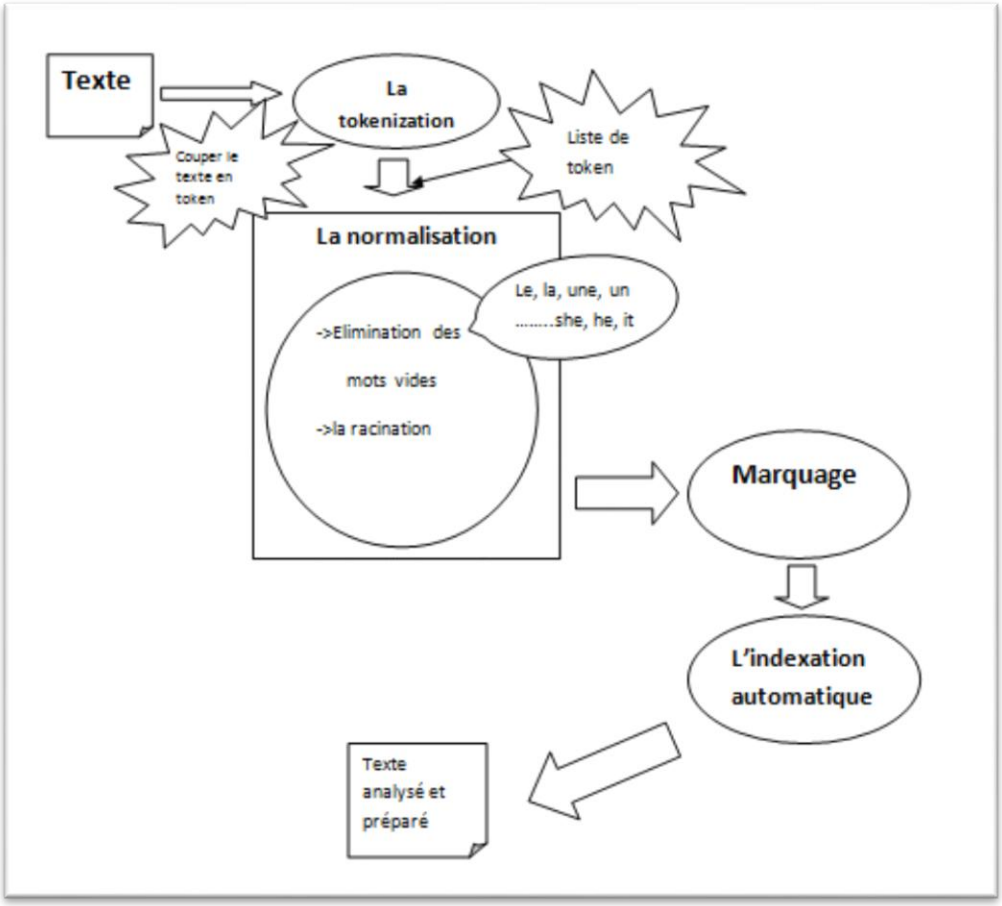


Figure 3.5 Processus d'analyse et préparation du texte

L'analyse consiste à préparer et filtrer le corpus pour extraire les termes importants du document. Cette analyse contient plusieurs étapes tel que la Tokenisation qui permet de convertir le document en un ensemble de termes (cette étape permet de mettre tous les mots en minuscules, élimine les apostrophes, les tirés et les points. . . .), et la suppression des mots vides « Stop words »non-significatifs tel que : les pronoms personnels, les prépositions . . . etc. L'étape de l'élimination des mots vides concerne l'emploi d'un anti-dictionnaire (aussi appelé stoplist) afin d'enlever les mots usuels. Les termes sélectionnés sont ensuite lemmatisés pour remplacer chaque mot par son lemme. La figure (3.5) présente un exemple sur les étapes qui se produisent lors de l'analyse du texte.

Marquage : Cette technique cherche à localiser et classer les entités nommées dans un texte en catégories prédéfinies telles que : les noms de personnes, organisations, lieux, dates, quantités, pourcentages, heures, email ….etc.

L'indexation automatique : L'indexation automatique est l'opération qui consiste à faire reconnaître par l'ordinateur des termes figurant dans le titre, le résumé, le texte complet.

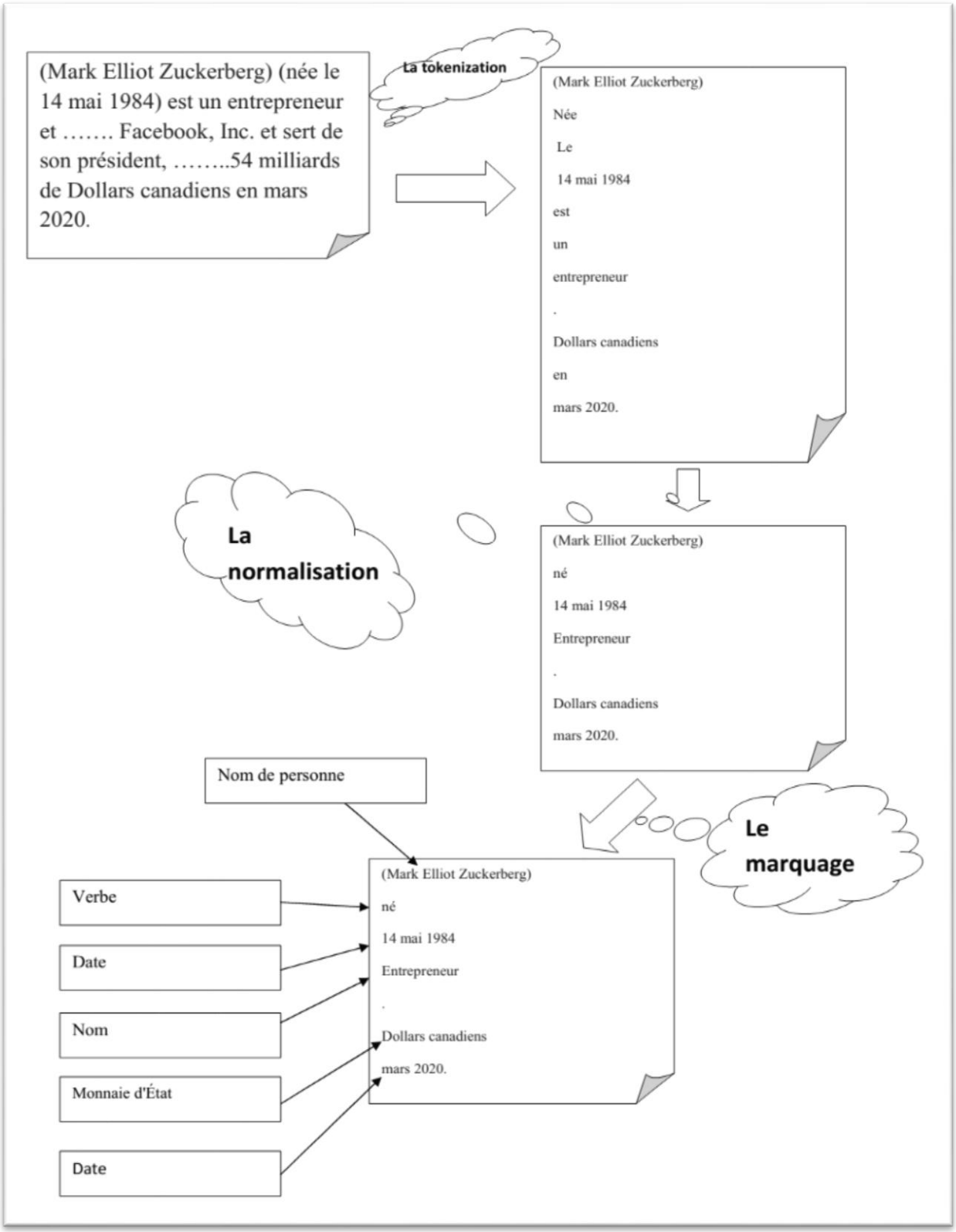


Figure 3.6 Exemple d'un texte analysé et pré-traité

3.3.4 Calcul de similarité entre concepts:

Le calcul de similarité sémantique est l'estimation d'un degré par lequel les concepts sont proches dans leur sens. Dans un paragraphe ou une phrase la similarité entre concepts des termes consiste à choisir une seule combinaison des sens (concepts) parmi les combinaisons possibles des concepts dans son contexte.

Exemple : prenons l'exemple de deux termes (computer et document).

Combinaison 1 : document#1 computer#1 = 0.2 (document#1 : le sens 1 du mot document")

Combinaison 2 : document#2 computer#1 = 0.4

Combinaison 3 : document#3 computer#1 = 0

Combinaison 4 : document#4 computer#1 = 0.82

Nous utilisons le contexte pour le calcul de similarité entre les concepts des termes extraits à partir du texte. Autrement dit, on détermine la meilleure combinaison dans laquelle les concepts sont sémantiquement très proches.

La proximité sémantique entre les concepts (sens) peut être évaluée par l'utilisation des mesures de similarités sémantiques (Wu-Palmer, la mesure de Rada et al, la mesure de Lin...). Ainsi, le concept sélectionné correspond au concept qui maximise la similarité sémantique avec les autres concepts du même contexte. Dans notre système nous basons principalement sur la mesure de (Wu-Palmer) et la ressource sémantique ontologie (WordNet) pour son calcul.

3.4 Modélisation Du Système :

Cette partie est consacrée à l'étape de modélisation, pour cela on va utiliser le formalisme d'UML (Unified Modeling Language) pour la modélisation de notre système.

1. Identification des acteurs : Les deux acteurs de notre système sont les suivants :

Administrateur : L'acteur principal dans notre système est l'administrateur. Un administrateur est l'expert de la base de documents qui dispose de toutes les fonctionnalités nécessaires à la gestion de la base de documents (ajout, suppression, mise à jour...).

L'administrateur de notre système assure les fonctionnalités suivantes :

-> S'authentifier.

->Gérer ses informations personnelles.

-> Gérer la base documentaire (Ajout, Suppression, Normalisation).

->Générer et consulter l'analyse.

Utilisateur : les tâches qu'un utilisateur de système peut réaliser sont :

->Inscription et authentification.

-> Gérer ses informations personnelles.

->Évaluation des résultats.

2. Identification des cas d'utilisations: Chacun des acteurs précédents à des tâches précises dans le système. Voici ce qu'associe à chaque acteur. Le diagramme de cas d'utilisation qui décrit les fonctionnalités de l'administrateur est présenté dans la figure3.6

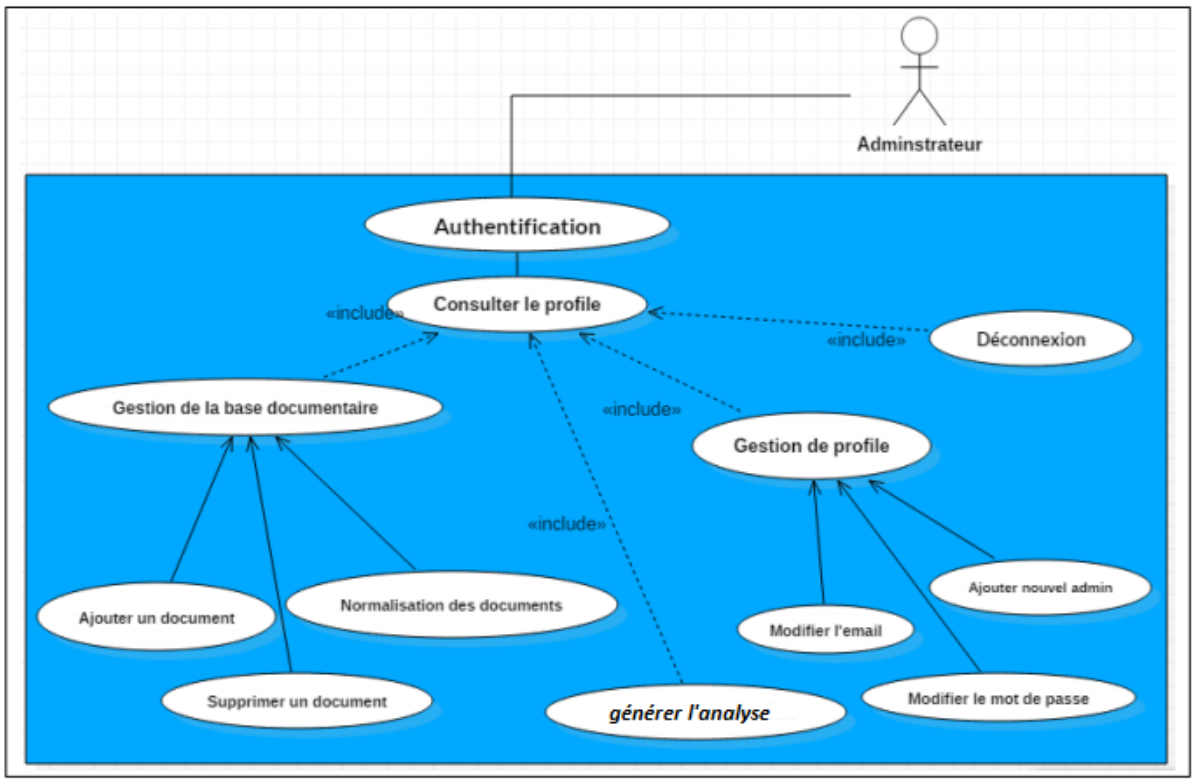


Figure 3.7 Diagramme de cas d'utilisation de l'administrateur du système

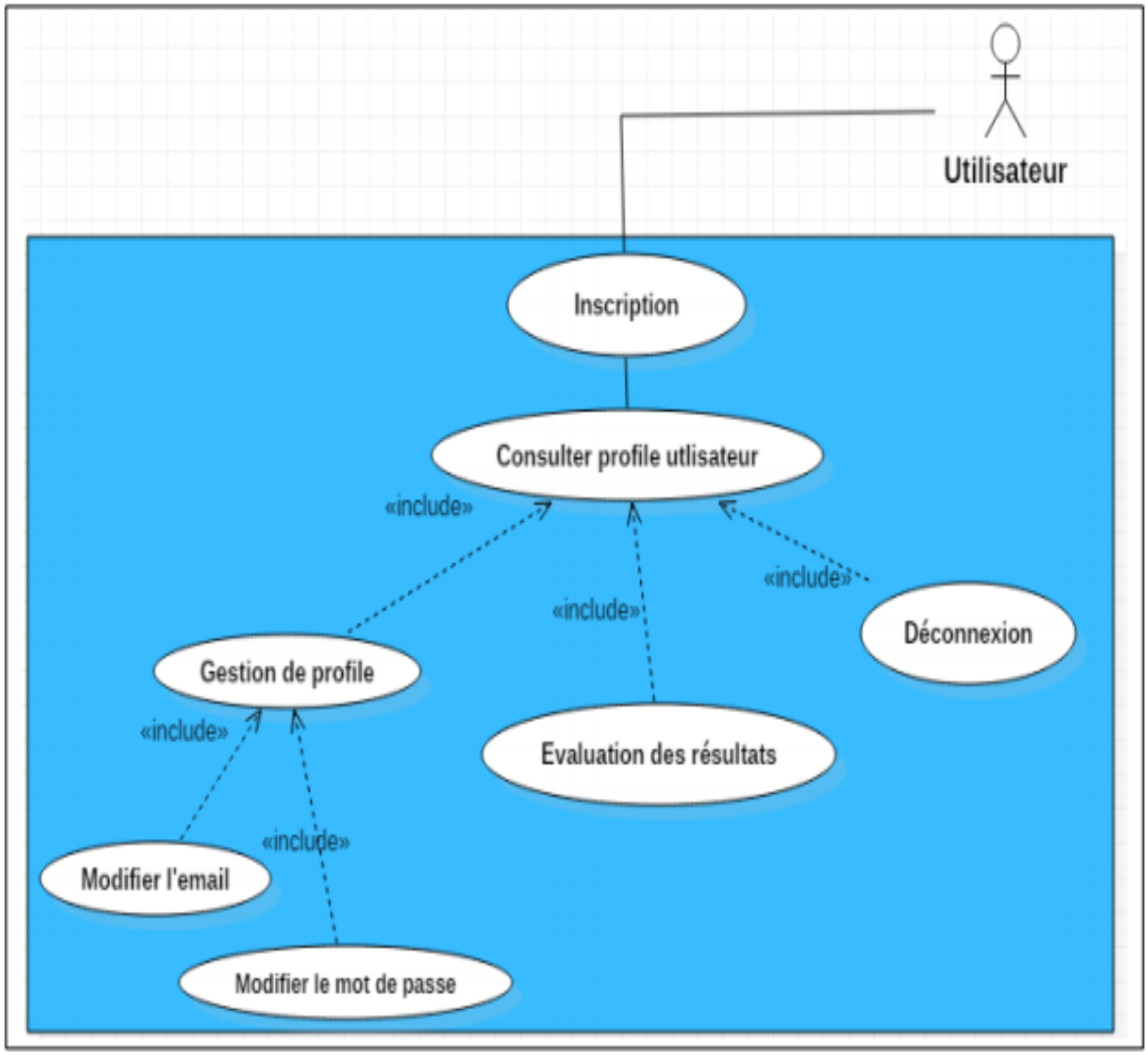


Figure 3. 8 Diagramme de cas d'utilisation de l'utilisateur du système

3.5 Conclusion :

Dans ce chapitre, nous avons décrit les besoins fonctionnels et techniques attendus de notre système. Nous avons détaillé le fonctionnement de tous les processus qui constituent ce système.

Dans le chapitre suivant nous allons présenter l'implémentation de tous les composants et les modules de notre système.

Chapitre4 :

Implémentation du système

4.1 Introduction :

Après avoir l'étude conceptuelle de notre approche d'extraction les concepts textuels, nous allons présenter dans ce chapitre la phase de réalisation et d'implémentation du notre système.

Ce chapitre a pour objectif de présenter l'aspect implémentation de notre application, il s'agit donc d'expliquer l'environnement matériel sur lequel notre système a été développé, les langages de programmation et les outils utilisés. Par la suite, nous allons présenter les interfaces graphiques en décrivant les différentes fonctionnalités de notre application et nous présenterons un exemple qui nous permettra d'illustrer les résultats obtenus lors de l'utilisation de notre approche.

4.2 Les Outils Et Librairies Utilisés:

Dans cette partie, nous allons présenter la définition du langage que nous allons utilisés dans l'implémentation et la réalisation de notre système.

4.2.1 Python :

Python est un langage de programmation, interprété car, avant de pouvoir les exécuter, un logiciel spécialisé se charge de transformer le code du programme en langage machine, multi -paradigme et multiplateformes, est placé sous une licence libre. qui vous permet de travailler rapidement et d'intégrer les systèmes plus efficacement. Python peut être utilisé pour gérer des données volumineuses et effectuer des calculs complexes. Il existe ce qu'on appelle des bibliothèques qui aident le développeur à travailler sur des projets particuliers. Plusieurs bibliothèques peuvent ainsi être installées pour, par exemple, développer des interfaces graphiques en Python.

Ce choix a été motivé par les raisons suivantes :

- L'une des principales langues parmi les langues appropriées pour la programmation de problèmes d'apprentissage profond.
- Il dispose un grand nombre de bibliothèques pour le traitement du langage naturel, telles que NLPnet, NLTK, … .
- Un langage simple, productif et utilisable dans presque tous les domaines et systèmes.

Figure 4.1 PYTHON

4.2.2 Natural Language Toolkit (NLTK) :

La boîte à outils en langage naturel (NLTK) est une plate-forme utilisée pour créer des programmes Python qui fonctionne avec des données de langage humain pour une application dans le traitement statistique du langage naturel (NLP).

Il contient des bibliothèques de traitement de texte pour la tokenisation, l'analyse, la classification, la racine, le marquage et le raisonnement sémantique. Il comprend également des démonstrations graphiques et des exemples d'ensembles de données, ainsi qu'un livre de recettes et un livre expliquant les principes sous-jacents des tâches de traitement du langage prises en charge par NLTK.

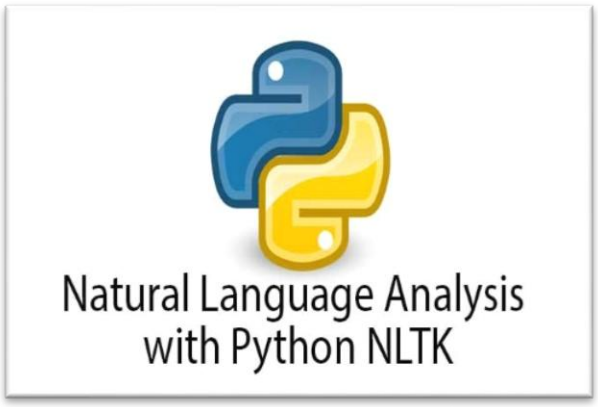


Figure 4.2 NLTK

4.2.3 Whoosh:

Whoosh est une bibliothèque de moteur de recherche Python pure et rapide. Le principal moteur de conception de Whoosh est qu'il s'agit de pur Python. Vous devriez pouvoir utiliser Whoosh partout où vous pouvez utiliser Python, aucun compilateur ou Java requis. Comme l'un de ses ancêtres, Lucene, Whoosh n'est pas vraiment un moteur de recherche, c'est une bibliothèque de programmation pour créer un moteur de recherche.

Figure 4.3 Whoosh

4.2.4 OS :

Ce module fournit une manière portable d'utiliser les fonctionnalités dépendantes du système d'exploitation. Si vous voulez uniquement lire ou écrire dans un fichier.

4.2.5 String :

Les chaînes peuvent être créées en insérant des caractères entre guillemets simples ou doubles. Même les guillemets triples peuvent être utilisés en Python mais généralement utilisés pour représenter des chaînes multilignes et des docstrings.

4.2.6 Hashedindex :

Implémentation rapide et simple d'InvertedIndex à l'aide de listes de hachage (dictionnaires python).

4.2.7 Sklearn :

Scikit-learn est une bibliothèque d'apprentissage automatique gratuite pour Python. Il comporte divers algorithmes tels que la machine vectorielle de support, les forêts aléatoires et les k-voisins, et il prend également en charge les bibliothèques numériques et scientifiques Python telles que NumPy et SciPy.

Figure 4.4 sklearn

4.2.7 NumPy :

NumPy est une bibliothèque python utilisée pour travailler avec des tableaux. Il a également des fonctions pour travailler dans le domaine de l'algèbre linéaire, de la transformée de Fourier et des matrices. NumPy a été créé en 2005 par Travis Oliphant. C'est un projet open source et vous pouvez l'utiliser librement.

Figure 4.5 NumPy

4.2.8 WordNet :

WordNet est une grande, base de données lexicale largement utilisée pour l'anglais. Ce lexique comporte environ 180000 termes organisés dans 117597 synsets qu'ils sont catégorisés en fonction de leurs catégories syntaxiques telles que le verbe, le nom, l'adjectif et l'adverbe. Son but est de répertorier, classifier et mettre en relation de diverses manières le contenu sémantique et lexical de la langue anglaise.

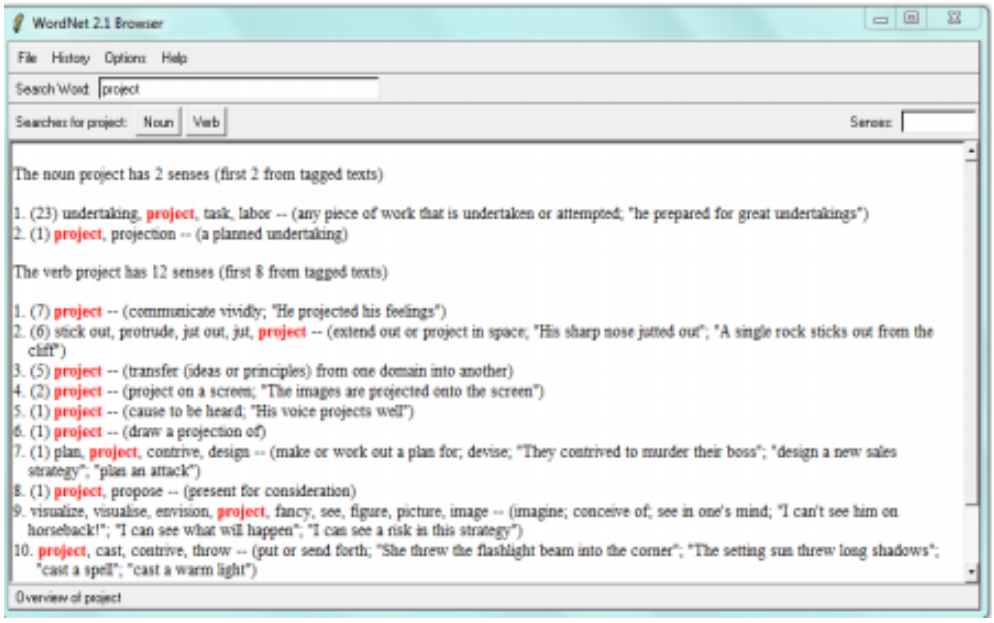


Figure 4.6 Interface du WordNet

4.2.9 Tkinter :

```
from tkinter import *
fen0 = Tk()
```

Figure 4.7 création frame

4.3 L'environnement de développement :

PyCharm est un environnement de développement intégré (IDE) utilisé dans la programmation informatique, spécifiquement pour le langage Python. Il est développé par la société tchèque JetBrains. Il fournit l'analyse de code, un débogueur graphique, un testeur d'unité intégré, l'intégration avec des systèmes de contrôle de version (VCS) et prend en charge le développement Web avec Django ainsi que la science des données avec Anaconda.

Figure 4.8 Pycharm

4.4 Implémentation :

4.4.1 Le prétraitement d'un document :

Le prétraitement est fait à l'aide la bibliothèque NLTK :

```
import nltk
from nltk.corpus import stopwords
```

La figure (ci-dessous) représente le processus de prétraitement d'un document avec quelques fonctions prédéfinies en python pour le traitement du langage naturel.

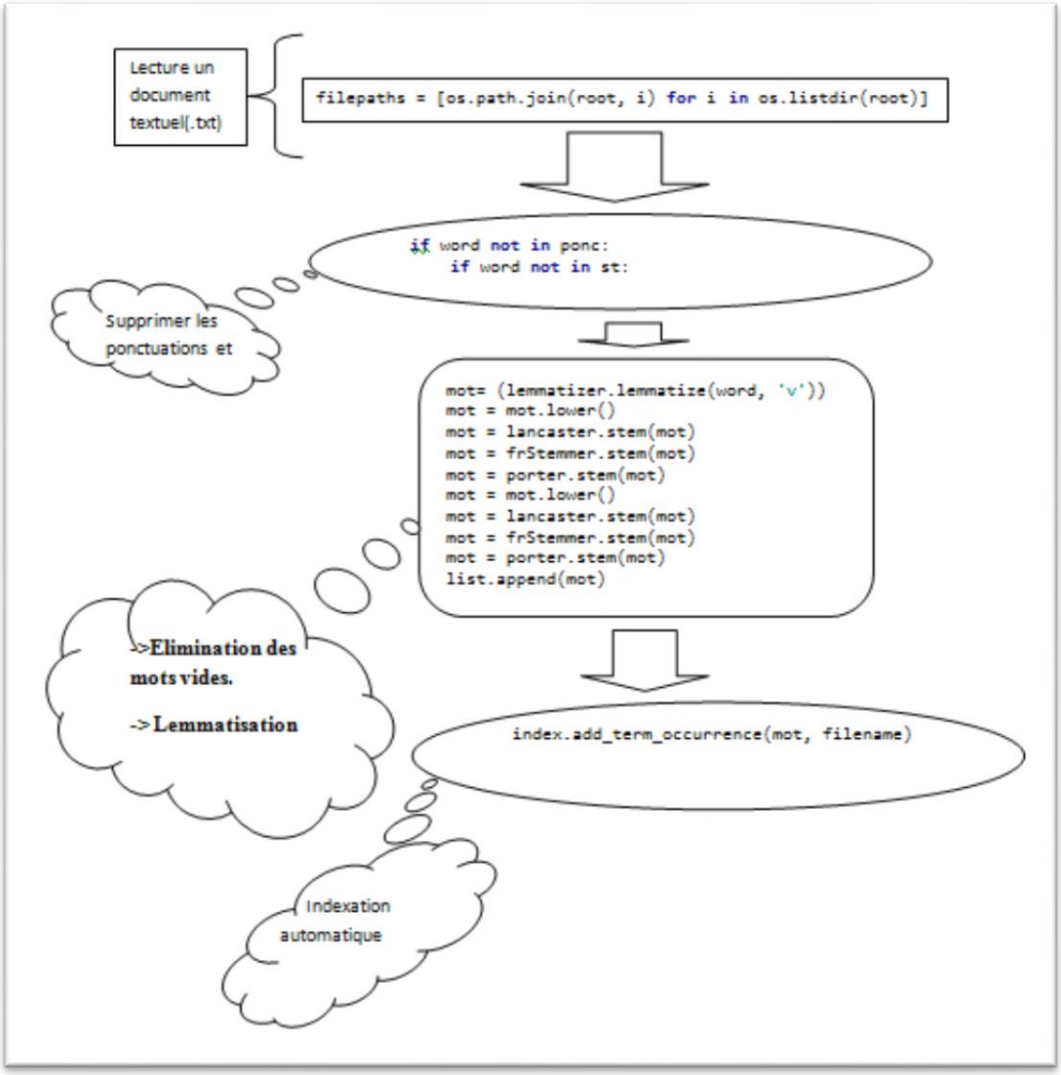


Figure 4.9 processus de l'exécution du prétraitement d'un document (.Txt)

4.4.2 Extraction Des Mots Composé Et Le Marquage :

Figure 4.10 processus d'extraction des mots composés et le marquage.

4.4.3 Calcule La Similarité :

```
import sklearn_features
import numpy as np
import nltk, string
from nltk import word_tokenize
from sklearn.feature_extraction.text import TfidfVectorizer
import stop_words
#calculer la sumilarité entre deux fichier
f1=open("f1.txt")
f2=open("f2.txt")
st = set(stop_words.get_stop_words('english'))
tfidf = TfidfVectorizer().fit_transform(f1,f2)
pairwise_similarity = tfidf * tfidf.T
print(pairwise_similarity)
```

Figure 4.11 Calcule la similarité

Résultat :

Figure 4.12 calcule la similarité

4.4.4 Présentation De La Fenêtre D'application :

Figure 4.13 Fenêtre1 d'application.

Commencer le programme

Figure 4.14 Fenêtre2 d'application.

Button analyse : pour faire analyse de texte

Button E_M_C : pour l'extraction les mots composés

Button C_S_S : pour calculer similarité sémantique

Button Quitter : pour sortir

4.5 Conclusion :

Les modèles actuels d'apprentissage profond reposent sur un bon traitement des données et sur le choix de l'algorithme approprié, mais les spécialistes ne sont pas encore parvenus à définir des paramètres statiques ni un modèle général permettant de résoudre tous les problème s similaires, mais ils ont atteint le soi-disant apprentissage par transfert. Ce chapitre , nous avons expliqué comment nous traitons des données textuelles et sélectionnons un modèle qui convient à nos données.

CONCLUSION GENERALE

Les textes non structurés sont des données représentées ou stockées sans format prédéfini. Cette absence de format entraîne des ambiguïtés qui peuvent rendre difficile la compréhension et l'exploitation de données.

Dans le cadre de notre travail , a passé en revue d'utilisation de l'apprentissage profond avec analyse de texte pour extraire les termes composés et les terme les plus importants des documents textuels, notamment les problèmes de langage naturel, la difficulté de les analyser et l'extraction du contenu le plus important des textes.

Les techniques d'apprentissage profonds ont été appliqués avec succès à de nombreuses tâches d'analyse de texte, conduisant à des systèmes efficaces, mais parfois à la taille du réseau (nombre accru de couches et de neurones) et au temps de formation sont interdits pour une utilisation efficace. Malgré cet inconvénient, il y a encore peu de travaux de recherche sur les moyens de trouver des moyens plus efficaces de former un réseaux de neurones profonds ou de trouver une structure optimale (sans former des centaines de réseaux différents)

Toutefois, le choix de l'extraction de certaines mots simples et composés dans notre prototype nécessite une analyse grammaticale du contenu pour détecter les portions du texte à utilisé, c'est pour cette raison nous avons utilisés le technique de deep learning.

BIBLIOGRAPHIE

[1] Elsa TOLONE. « Analyse syntaxique à l'aide des tables du Lexique-Grammaire du français ». Thèse de Doctorat, Université Paris-Est, mars 2011.

[2] Intelligence artificielle [cs.AI]. Université d'Avignon, 2017. Killian Janod.

[3] Using Convolutional Neural Networks to Extract Keywords and Keyphrases About Foodborne Illnesses by Jingjing Wang A Thesis presented to The University of Guelph.

[4] MATALLAH H, 2011.Classification Automatique de Textes Approche Orientée Agent, Mémoire de Magister. Université Aboubekr BELKAID-Tlemcen.

[5] Silvia F., Eric S., Juan M., Torres M., 2007. Énergie textuelle de mémoires associatives.

[6] Alexis C., Holger S., Yann Le Cun, 2016. Very Deep Convolutional Networks for Natural Language Processing.

[7] Vincent Bouchet, 2017. Mémoire de Master Machine learning en fnance.

[8] Jiwei L. , Xinlei C., Eduard H., Dan J., Visualizing and Understanding Neural Models in NLP

[9] https://www.tutorialspoint.com/

[10] JOINT-INTL.COM

[11] : https://www.reputationvip.com/fr/blog/

[12] N'TECHOBO Edoukou Philippe Armel. « Annotation sémantique et analyse de surface pour l'extraction de graphes d'abstraction de débats politiques ». Thèse de Doctorat, Université de Montréal, Juillet 2016

[14] Djamel NESSAH. « Un modèle de raisonnement pour un système de recherche sémantique d'informations sur le web basé agents ». Thèse de Doctorat en Informatique, Université de Mohamed kheider Biskra, 2014.

[15] Authoul Abdul Hay. « Constitution d'une ressource sémantique arabe à partir de corpus multilingues alignés ». Thèse de doctorat, Université Grenoble INP, novembre 2012

[16] Information Resources Management Association. « Information Retrieval and Management : Concepts, Methodologies, Tools, and Applications ». USA, Édition : 4, Février 2018.

[17] Gizem Sogancoglu, Hakime Ozturk, Arzucan Ozgu. « A semantic sentence similarity estimation system for the biomedical domain ». Bioinformatics, Volume 33, Issue 14, 15 Juil 2017, Pages i49-i58.

[18] Ahmad Fayez, S. Althobaiti . « Comparison of Ontology-Based Semantic Similarity Measures in the Biomedical Text ». Journal of Computer and Communications, 5, 17-27, 9 Fev 2017.

[19] Farah HARRATHI. « Extraction de concepts et de relations entre concepts à partir des documents multilingues : Approche statistique et ontologique ». Thèse de Doctorat, INSA de Lyon, Septembre 2009.

[20] L. Moncla, M. Gaio. « Services Web pour l'annotation sémantique d'information spatiale à partir de corpus textuels ». SAGEO Spatial Analysis and GEOmatics 2017, Nov 2017, Rouen, France.

[21] Mohamed GASMI. « Raisonnement pour les logiques de description appliqué au web sémantique ». Thèse de Doctorat, Université de M'Sila, 2017.

[22] Iddir OUNNACI. « Recherche d'information dans les documents pédagogiques structurés adaptée aux besoins spécifiques des apprenants ». Mémoire de Magister en informatique, Université Mouloud Mammeri de Tizi-Ouzou. Février 2015.

[23] Badr-Eddine BENAISSA. « Construction semi-automatique d'ontologies à partir de textes arabes ». Mémoire de Magister, Université de Tlemcen, 2012.

[24] Université Paris 5 - Maîtrise de mathématiques - Maîtrise MASS – MST ISASH [L'INTELLIGENCE ARTIFICIELLE DEFINITION - GENERALITES - HISTORIQUE – DOMAINES].

[25] Les Cahiers Lysias [Intelligence Artificielle].

[26] [L`apprentissage profond (Deep Learning) pour la classification et la recherche d'images par le contenu] UNIVERSITE KASDI MERBAH OUARGLA Faculté des Nouvelles Technologies de l'Information et de la Communication Département d'Informatique et des Technologies de l'information.

[27] Introduction à l'apprentissage automatique MONOGRAPHIE DE CPA NOUVEAU-BRUNSWICK.

[28] L'ANALYSE DU SENTIMENT UTILISANT LE DEEP LEARNING Présenté par : Medjdoubi Abdelkader Encadré par : Dr. Yahlali Mebarka Dr. Boudia Mohamed Amine

[29] Text feature extraction based on deep learning [Lianget al. EURASIP Journal on Wireless Communications and Networking (2017) 2017:211]

[30] De https://www.saagie.com/fr/blog/qu-est-ce-que-le-deep-learning/

[31] : Taweh Beysolow, Applied Natural Language Processing with Python Implementing Machine Learning and Deep Learning Algorithms for Natural Language Processing

[32] De http://www.fullai.org/short-history-artificial-intelligence/.

[33] De http://blogshells.com/how-deep-learning-and-data-science-work-with-natural-languageprocessing/

[34] : Deep Neural Networks for Text: A Review Chiung Ching Ho1 , Khairul Nizam Baharim2 , Ahmad Abdulsalam Ahmad Fatan2 , and Mohd Shafiq Bin Alias2 1Multimedia University, Data Science Institute, 2TM Research & Development Corresponding author's email : ccho@mmu.edu.my {khairulnizam,abdulsalam,shafiq}@tmrnd.com.my

[35]: Understanding LSTM – a tutorial into Long Short-Term Memory Recurrent Neural Networks.

[36] Akshay K.,Adarsha S., Natural Language Processing Recipes Unlocking Text Data with Machine Learning and Deep Learning using Python