REPUBLIQUE ALGERIENGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE DE KHENCHELA
FACULTE DES MATHEMATIQUES ET DE L'INFORMATIQUE
Département de L'informatique
Mémoire de fin d'étude
Présenté Pour l'obtention Du diplôme de Master

Domaine : Mathématiques et Informatique
Filière : Informatique
Spécialité : Sécurité et Technologie de Web (STW)

Par : Guerrab arifa
Rezeimia linda

THEME

WORD SENSE DESAMBIGUATIONBY HIGH UTILITY PATTERNS IN SENTENCES

Soutenu le : 27/06/2022

Devant le jury composé de :

- Dr.kheiar	**Université de Khenchela**	**Président**
- Dr.Ledmi Makhlouf	**Université de Khenchela**	**Rapporteur**
- Dr.Bakhoche Abdelali	**Université de Khenchela**	**Co-Rapporteur**
- Dr.Rehabi hichem	**Université de Khenchela**	**Examinateur**

Promotion : **2021/2022**


بسم الله الرحمن الرحيم

-بسم الله الرحمان الرحيم -

صدق الله العظيم

Dédicaces

À mes parents ;

Mon père Hawasse Rezaimia , le plus beau père Qui ne m'a rien épargné.

Ma mère,Lwiza la plus belle mère que je ne suis rien sans . Qu'Allah les garde en bonne santé !

À mon cher mari bilal, il a toujours été un soutien pour moi, Aux deux perles de ma vie : Mes deux fils, Nourssin,Soujod, Qu' Dieu les protège !

À ma chère sœur wafa et mes deux frères abd sslam ,Roro sont mon soutien et ma force.

A mon oncle chemchar Rebai , que Dieu lui fasse miséricorde et lui accorde le paradis A ma tante 3akila et son fils bichou et ses filles ibtessem,chahinez

A ma grand-mère et mon oncle hafsi que Dieu lui donne longue vie

À toutes les personnes qui m'aiment qu'ils trouvent

Dédicaces

Guerrab arifa

Dieu merci, j'ai atteint ce stade et réalisé mon souhait et les souhaits
de tous ceux qui m'aiment et m'aident

Je dédie cet humble voyage
ceux qui ont fait de moi une femme, très chers
Mon père est "mon père" et "ma mère".
Ma petite famille et mes enfants sont les rayons du soleil originel, la lumière de la certitude et la lumière de la foi, et j'honore mon cher fils, mon Seigneur,
protège -les, si Dieu le veut, et mon cher mari qui s'est tenu à mes côtés pour achever mon parcours universitaire, et mon oncle et sa femme la religion
m'ont soutenu dans ce voyage. C'est le voyage, et je remercie beaucoup ma cousine, Nafisa, qui s'est inscrite et m'a aidé à terminer mon parcours universitaire. Que Dieu lui accorde ce que nous souhaitons
A mes frères et sœurs et à toute la famille, et je dis à ma sœur aînée, Rabia,
que j'ai réalisé votre souhait. Vous avez toujours souhaité que je sois ingénieur
Ici, je réalise mon souhait. Je remercie tous ceux qui ont soutenu m'a aidé
à terminer mon parcours universitaire. Punition pour tous les garçons et les filles.
En savoir plus sur ce texte source Vous devez indiquer le texte source pour obtenir des
Informations supplémentaires
Envoyer des commentaires
Panneaux latéraux

.Remerciements

*G*râce à Allah tout d'abord de me donner la puissance, la santé et l'aide d'accomplir cette
Recherche. Sans l'aide de Dieu, je ne pouvais pas accomplir ce travail.
*J*e souhaite ensuite exprimer à monsieur **ledmi makhlouf ,** qui a dirigé mes
Travaux, ma plus profonde gratitude pour sa disponibilité, ses conseils clairvoyants, son
Soutien sans faille et la confiance qu'il a bien voulu m'accorder.
*J*e souhaite aussi témoigner de ma sympathie et de ma gratitude à tous ceux qui ont
Toujours été agréable avec moi
*M*es plus affectueux remerciements vont évidemment à toute ma famille et tout d'abord à
Mes parents qui m'ont toujours soutenu et encouragé dans tout ce que j'ai entrepris.
*M*es remerciements vont aussi à toute l'administration de Département de l'informatique, pour leur gentillesse
*Q*ue toutes les personnes qui ont attribué de près ou de loin à l'élaboration de ce travail

*M*erci à tous et à toutes.

Sommaire

Tableau de figure

Liste des Tableaux

Introduction générale

Introduction générale

Dans la linguistique computationnelle, la désambiguïsation du sens du mot est un problème ouvert du traitement du langage naturel et de l'ontologie, qui régit le processus d'identification du sens d'un mot utilisé dans une phrase, lorsque le mot a plusieurs significations. La solution à ce problème affecte les autres écritures informatiques, telles que le discours, l'amélioration de la pertinence des moteurs de recherche, la résolution anaphore, la cohérence, l'inférence et autres. La recherche a progressivement progressé jusqu'à ce que les systèmes WSD atteignent des niveaux de précision suffisamment élevés sur divers types de mots et ambiguïtés. Une variété riche de techniques a été recherchée, à partir de méthodes basées sur le dictionnaire qui utilisent les connaissances encodées dans les ressources lexicales, aux méthodes d'apprentissage par les approches d'apprentissage supervisées ont été les algorithmes les plus réussis à ce jour. La précision actuelle est difficile à déclarer sans une série de mises en garde.

RESUME

Résumé : La Fouille de motifs fréquents vise à identifier des récurrences, que l'on appelle motifs fréquents, parmi les enregistrements dans les bases de données. Ces motifs fréquents peuvent être des ensembles d'items fréquents, des sous-séquences fréquentes ou tout autre type de sous-structures comme les graphes, les arbres, … etc

De plus, la fouille des motifs à haute utilité est une tâche émergente de la fouille des données, qui consiste à découvrir des motifs ayant une grande importance dans les bases de données. L'utilité d'un motif peut être mesurée en fonction de divers critères objectifs tels que son profit, sa fréquence et son poids.

D'autre part, la désambiguïsation permet d'améliorer de nombreuses applications en traitement automatique des langues (TAL) comme la recherche d'information, l'extraction d'information, la traduction automatique, ou la simplification lexicale de textes.

Schématiquement, il s'agit de choisir quel est le sens le plus approprié pour chaque mot d'un texte.

Dans ce sujet, nous proposons d'appliquer l'une des méthodes de la fouille des motifs à haute utilité pour la désambiguïsation des mots dans un corpus textuel.

Mots-clés : WSD, traitement du langage naturel, méthodes d'apprentissage.

Abstract: Frequent Pattern Mining aims to identify recurrences, called frequent patterns, among records in databases. These frequent patterns can be sets of frequent items, frequent sub-sequences or any other type of sub-structures such as graphs, trees, etc.

Moreover, high-utility pattern mining is an emergent task of data mining, which consists in discovering patterns with high importance in databases. The usefulness of a pattern can be measured based on various objective criteria such as its profit, frequency, and weight.

On the other hand, disambiguation improves many applications in automatic language processing (TAL) such as information retrieval, information extraction, machine translation, or lexical simplification of texts. Schematically, it is a question of choosing what is the most appropriate meaning for each word of a text.

In this topic, we propose to apply one of the highly useful pattern mining methods for word disambiguation in a textual corpus.

الخلاصة:

يهدف التعدين المتكرر للأنماط إلى تحديد التكرارات، التي تسمى الأنماط المتكررة، بين السجلات في قواعد البيانات. يمكن أن تكون هذه الأنماط المتكررة مجموعات من العناصر المتكررة أو التسلسلات الفرعية المتكررة أو أي نوع آخر من الهياكل الفرعية مثل الرسوم البيانية والأشجار وما إلى ذلك.

علاوة على ذلك ، يعد التعدين بنمط المنفعة العالية مهمة ناشئة للتنقيب عن البيانات ، والتي تتمثل في اكتشاف أنماط ذات أهمية عالية في قواعد البيانات. يمكن قياس فائدة النمط بناءً على معايير موضوعية مختلفة مثل ربحه وتكراره ووزنه.

من ناحية أخرى، يحسن توضيح العديد من التطبيقات في المعالجة التلقائية للغة (NLP) مثل استرجاع المعلومات أو استخراج المعلومات أو الترجمة الآلية أو التبسيط المفرد للنصوص. من الناحية التخطيطية، يتعلق الأمر باختيار المعنى الأنسب لكل كلمة في النص.

Problématique

La désambiguïsation du sens des mots, dans le traitement du langage naturel (TALN), peut être définie comme la capacité à déterminer quel sens du mot est activé par l'utilisation du mot dans un contexte particulier. L'ambiguïté lexicale, syntaxique ou sémantique, est l'un des tous premiers problèmes auquel est confronté tout système TAL. Les marqueurs de partie du discours (POS) avec un haut niveau de précision peuvent résoudre l'ambiguïté syntaxique de Word. D'un autre côté, le problème de la résolution de l'ambiguïté sémantique est appelé DSM (word sense disambiguation). Résoudre l'ambiguïté sémantique est plus difficile que résoudre l'ambiguïté syntaxique.

Objectifs

L'objectif global de notre travail, est d'offrir une méthode automatisée pour la désambiguïsation du sens des mots, pour ce faire, différents axes d'étude ont été envisagés :
Le premier axe combine traitements automatique du langage naturel. Nous avons appliqué des prétraitements linguistiques au corpus dans le but d'améliorer la représentation des textes.

La deuxième axe permet de choisir des méthodes et des algorithmes pour determiner le sens exact de mot.

Organisation du mémoire

- **Dans le premier chapitre,** Nous présentons deux sections principales. La première section du chapitre aborde les définitions et le processus de la fouille de données, les différents types et les taches de cette fouille de données. La deuxième section détaille cette technologie appliquée à la donnée textuelle, le fouille de texte. Après l'exposition des définitions et des approches de la fouille de texte, nous expliquons la chaine de traitement pour le processus de fouille de données textuelle. Puis nous citons quelques applications pour cette technologie.
- **Dans le deuxième chapitre**, nous présentons deux sections très importantes dans ce travail, une partie pour le traitement du langage naturel on présenté ces approches et ses application, la deuxième section traite la tâche de désambiguïsation du sens des mots, DSM est les plus difficiles problèmes ouverts de la TAL. Certains des défis du DSM ont été discutés, de nombreux problèmes surviennent en DSM parce qu'elle dépend de connaissances tirées de différentes ressources.
- **Dans le troisième chapitre** nous présenterons la méthode que nous avons proposée pour la désambiguïsation du sens des mots (DSM), ainsi que les outils utilisés pour le développement du system tels que le choix du langage, l'environnement, ainsi que l'ensemble des résultats des expérimentations obtenus.

Chapitre01 :
Fouille du Texte

Chapitre 1 Fouille De Texte

Introduction

Dans ce chapitre Nous présentons deux sections principales. La première section Aborde les définitions et les processus de la fouille de données, l'importance de l'utilisation, les différents types et les taches de cette fouille de données. La deuxième section détaille cette technologie appliquée aux données textuelles, nous avons donné les définitions et les approches du Fouille De Texte, la chaine de traitement pour le processus de fouille de données textuelle et la fouille des items fréquents.

1- Fouille de donnes

1-1- Définition

La fouille de donnes est définie comme un processus utilisé pour extraire des données utilisables d'un ensemble plus large de données brutes. Le Fouille de donnes implique la collecte et l'entreposage efficaces des données ainsi que le traitement informatique. Pour segmenter les données et évaluer la probabilité d'événements futurs, le Fouille de donnes utilise des algorithmes mathématiques sophistiqués[1].

1-2- Processus du Fouille de donnes

Cross Industry Standard Processus for Fouille de donnes (CRISP-DM) est un modèle de processus qui sert de base à un processus de science des données.

Il comporte six phases séquentielles :

[1] https://www.oracle.com/fr/database/data-mining-definition.html

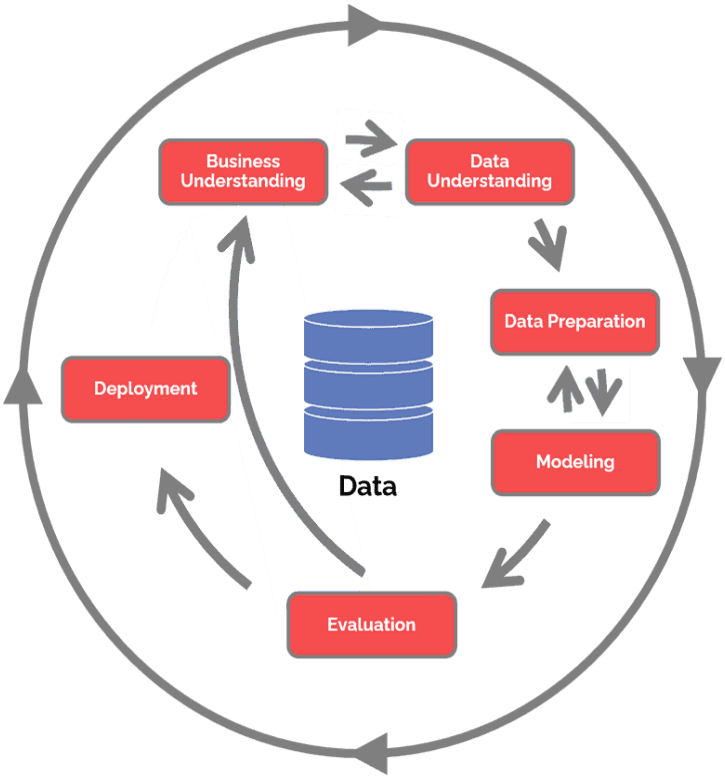


Figure 1.1**:** CRISP-DM Diagramme.[2]

1. Compréhension du problème (phase de compréhension commerciale)

2. Compréhension des données (phase de compréhension des données)

3. Préparation des données (phase de préparation des données)

4. Modélisation (phase de modélisation)

5.Évaluation (phase d'évaluation d'étape)

6. Déploiement : mettre les résultats de l'analyse à la disposition des décideurs et utiliser les informations finales pour adapter la stratégie.[3]

1-3- Comment faire la fouille de donnes ?

La fouille de donnes est majoritairement utilisée dans les domaines de **l'analyse de la consommation et de la relation client.** La fouille de données liées aux

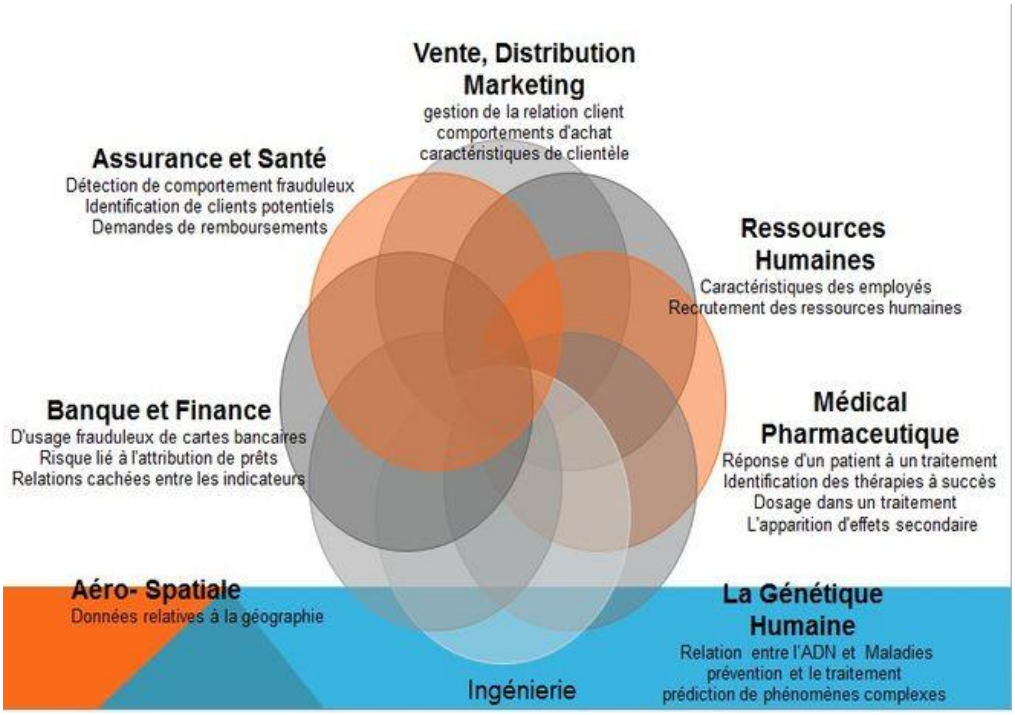
[2] https://www.datascience-pm.com/crisp-dm-2/
[3] https://www.talend.com/resources/what-is-data-mining/

comportements des consommateurs permet d'optimiser l'offre commerciale et l'expérience client, de gagner en efficacité dans la stratégie marketing et d'améliorer l'image de marque de l'entreprise. Par exemple du succès de la fouille de donnes : le système avancé d'exploration de données de Netflix permet à l'entreprise de proposer des suggestions personnalisées afin d'améliorer l'expérience utilisateur ainsi que l'image de marque innovante de l'entreprise.

En banque, la fouille de donnes est utilisée pour scorer les clients et les classer en fonction de leur niveau de risque. De cette manière, l'établissement de crédit est en mesure d'adapter sa politique commerciale de manière sécurisée. La banque par exemple exige des garanties supplémentaires pour accorder un prêt à un client risqué. La fouille de données en matière bancaire est également utile pour la détection des fraudes.

En vente par correspondance les sociétés de vente par correspondance ont recours à l'exploration de données pour identifier le profil de ce type de consommateurs, de manière à axer leurs actions marketing et commerciales sur cette cible de clientèle, pour in fine optimiser leurs coûts[4].

1-4- Domaine d'application



[4] https://blog.hubspot.fr/marketing/data-mining

Figure 1.2 : Les domaines d'application[5].

1-5- Principales tâches de la fouille de donnes

Les tâches les plus courantes que le fouille de donnes est amené à accomplir :

- La description
- L'estimation
- La prévision
- La classification
- Le clustering
- L'association

1-5-1 La description :

Parfois, les chercheurs et les analystes essaient simplement de trouver des façons de décrire des tendances cachées dans les données. Les descriptions des modèles et des tendances servent à expliquer ou vérifier un fait.

1-5-2 L'estimation :

L'estimation est similaire à la classification, sauf que la variable cible est numérique plutôt que catégorique. Les modèles sont construits en utilisant des données, qui fournissent la valeur de la variable cible, ainsi que les « prédicateurs ». Par exemple : « l'estimation de la pression artérielle d'un patient d'hôpital, basée sur son âge, son sexe, son indice de masse corporelle, et le taux de sodium. La relation entre la pression artérielle et le prédicteur variable de l'ensemble de formation nous donnerait un modèle d'estimation. Nous pouvons alors appliquer ce modèle à de nouveaux cas.

1-5-3 La prédiction:

La prédiction est semblable à la classification et l'estimation, sauf que pour la prévision, les résultats se situent dans l'avenir. Exemples de tâches de prévision appliquée au marketing : « Prédire le prix d'un stock de trois mois dans le futur »

1-5-4 La classification :

[5] https://fr.wikiversity.org/wiki/Datamining/Applications

Supposons qu'un décideur veuille classer ses employés par tranches de revenu, ou n'importe quelle autre caractéristique associée à cette personne, comme l'âge, le sexe et la profession. Cette tâche est une tâche de classification.

1-5-5 Le clustering :

Le clustering désigne le regroupement des données, des observations ou des cas dans des classes d'objets similaires. Un cluster maximise la similarité des objets de du même cluster et minimise la similarité des objets de cluster différents. En effet,. La tâche de clustering ne cherche pas à classer, estimer, ou prédire la valeur d'une variable cible. Mais plutôt à segmenter l'ensemble des données en sous-groupes relativement homogènes à l'aide de mesures de distances.

1-5-6 L'association :

La recherche de règles d'association est la tâche la plus intéressante de la fouille de donnes. C'est également celle qui est la plus répandue dans le monde des affaires, notamment en marketing pour l'analyse du panier de consommation. La recherche de règles d'association cherche à découvrir les règles de quantification ou de relation entre deux ou plusieurs attributs. Les règles d'association sont de la forme «Si antécédent, puis conséquente », avec une mesure confiance associée à la règle. La recherche de règles d'associations dans une grande base de données permet de découvrir des règles cachées utiles pour la prise de décision[6].

2- Fouille de textes (Text Mining)

2-1- Fouille de texte

Historiquement la fouille de donnes est à la base du Fouille de texte au sens où celui-ci est l'extension du même but et du même processus vers des données textuelles. Néanmoins, les deux technologies se distinguent dans la nature des données à traiter. La fouille de donnes s'intéresse aux données numériques et factuelles qui sont bien structurées dans des bases de données, alors que la fouille de texte s'intéresse aux données textuelles non structurées, généralement exprimées en langage naturel. (Houria, 2012)

2-2- Définition

[6] https://www.petite-entreprise.net/P-2595-83-G1-principales-taches-du-data-mining.html

Le Fouille de texte est une technique permettant d'automatiser le traitement de gros volumes de contenus texte pour en extraire les principales tendances et répertorier de manière statique les différents sujet évoqués Les techniques de fouille de texte sont surtout utilisées pour des données déjà disponibles au format numérique. (Bathelot, 2017)

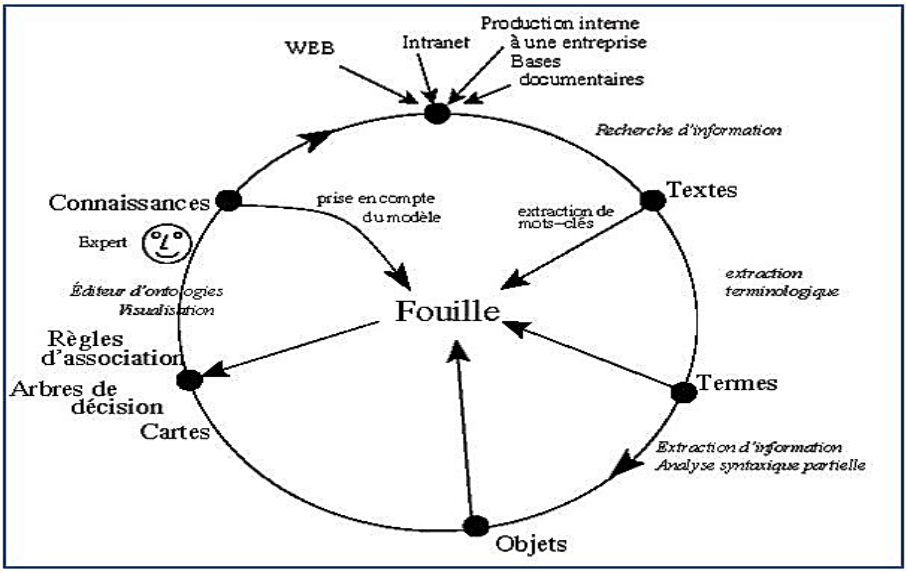


Figure 1.3 **:** Définition de fouille de texte (Bathelot, 2017)

Le text mining est donc : procédé consistant a synthéser (classer,structurer ,résumer …) les textes en analysant les relation ,les patterns,et les régles entre unités textuelles (mots,groupes,phrases,documents).

2-3- Techniques

- Classification
- Apprentissage
- Recherche d information
- Statistiques
- Extraction de patterns et d'entités
- raisonnement basé cas
- TALN=technique danalyse du langage naturel

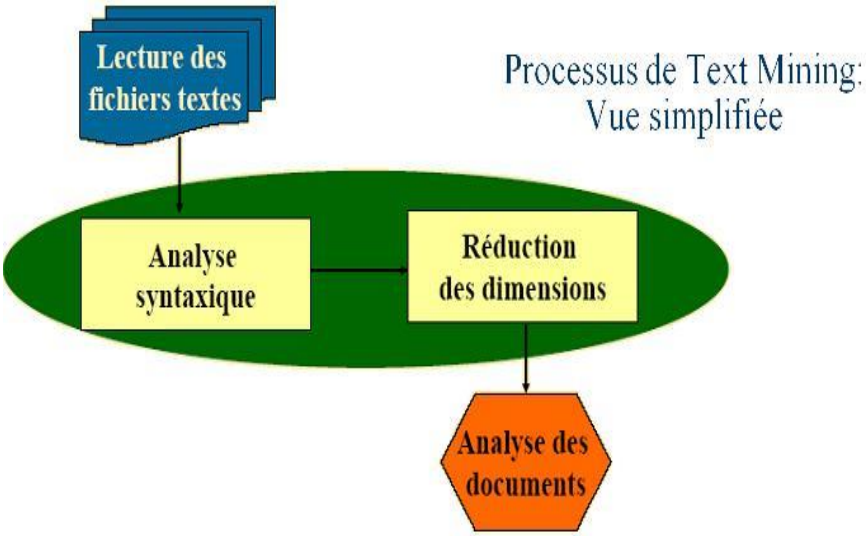


Figure1.4**. :** Vue simplifiée de processus de fouille de texte (Text Mining) [7]

2-4- Comment fonctionne le fouille de texte (text mining) ?

Les outils de fouille de texte doivent notamment respecter quelques règles de base dans leur traitement. Ces règles de base sont généralement et chronologiquement les suivantes :

- Tout d'abord, le logiciel de fouille de texte doit être en mesure de reconnaître les unités de la langue, en d'autres termes les mots (tokenisation);
- Par la suite, ce même logiciel de, fouille de texte doit réussir à interpréter la ponctuation et la mise en page des documents analysés (paragraphes, retours à la ligne, etc);
- Les formes grammaticales et lexicales doivent elles aussi être prises en compte dans l'analyse des corpus de textes. A noter que ces formes sont amenées à énormément varier selon les langues (anglais, arabe, chinois…..);
- L'outil de fouille de texte doit ensuite respecter une phase de lemmatisation consistant à identifier les différentes déclinaisons ou flexions d'un terme

L'ensemble de ces phases précédemment décrites relèvent de l'analyse linguistique permettant aux outils de fouille de texte d'établir un document transformé ; le document initial étant fait pour être lu par des yeux humains, le document après traitement étant destiné aux machines Deux approches non antinomiques sont par la suite envisagées :

une approche statistique et une approche sémantique.

[7] https://touriaelouahabi.wordpress.com/text-mining/definition-du-text-mining/

- **L'approche statistique du fouille de texte (text mining) :**

  Cette approche consiste à ne percevoir le document traité que via le prisme des chiffres. De cette manière, l'outil statistique de fouille de texte engendre des informations portant sur le nombre d'occurrence d'un terme, de cooccurrence de plusieurs termes ainsi que la fréquence d'apparition d'un terme dans un document ou corpus de textes.

  L'approche statistique du fouille de texte peut également produire des vecteurs de sens, pouvant être définis comme des statistiques de cooccurrence de termes, permettant de classer et/ou catégoriser un ensemble de textes dans un corpus.

- **L'approche sémantique du fouille de texte (text mining) :**

  L'approche sémantique du fouille de texte se base non plus sur la puissance de calcul, mais sur un élément externe, appelé le référentiel. Les référentiels peuvent être des listes à plats, des mots clés, des ontologies ou bien des thesaurus (liste organisée de termes normalisés). Ces référentiels peuvent également être des logiques de type probabilistes, tels que les réseaux bayésiens notamment utilisés pour la détection de spams et le fouille de donnes. Le moteur de fouille de texte va alors ajouter au document traité l'ensemble des informations fournies par un référentiel. Le référentiel effectue donc un travail de déduction avant de fournir au moteur de fouille de texte une réponse venant enrichir le document traité.

  Les avantages de l'approche sémantique du fouille de texte résident dans les paramètres du moteur de fouille de texte, qui peuvent être ajustés de manière à coller à la spécificité du corpus documentaire exploité. Il est également possible de modéliser des connaissances métiers spécifiques, de manière à effectuer des traitements fouille de texte répondant à des besoins bien spécifiques. La pertinence des résultats obtenus via une approche sémantique de la fouille de texte est généralement plus fine que celle obtenue par une approche statistique.[8]

2-5- Chaîne de traitement pour le processus de Fouille de texte

Nous décrivons le processus de par la **figure 1.5** et montre les différentes étapes de traitement dans un processus de FdT. Les données traitées sont constituées d'un ensemble de textes. Chaque texte est représenté par un ensemble de mots-clés. Cette représentation est stockée dans une base de données. Nous considérons un texte comme une entité porteuse

[8] https://ia-data-analytics.fr/logiciel-data-mining/text-mining/definition/

d'une information qu'il faut préparer, représenter et organiser pour que nous puissions utiliser des outils de fouille de données et valider les résultats de la fouille. La transformation des données textuelles en connaissances se compose donc de trois principales étapes :

1-La modélisation du contenu des textes
2- Les outils de fouille de données proprement dits
3-Le module d'analyse des résultats et leur validation

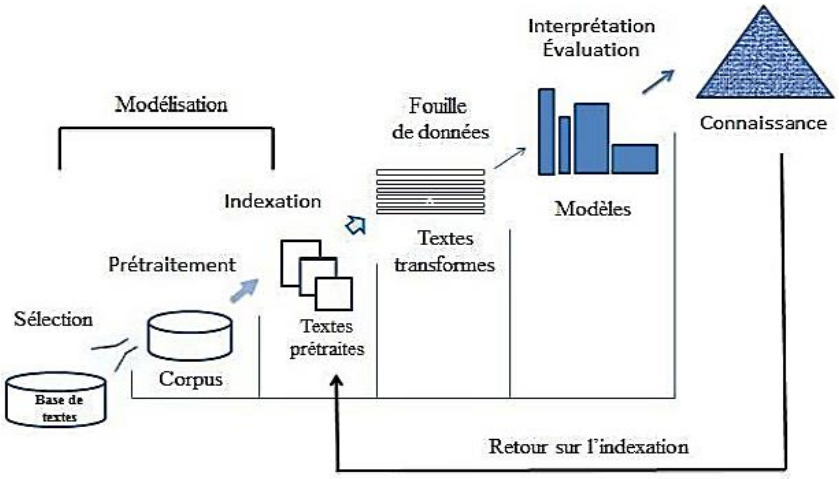


Figure 1.5 : La chaîne de traitement pour le processus de fouille de texte
(H. Cherfi, 2014)

2-6- Tâches principales de la fouille de textes

Dans cette section, nous allons énumérer les trois principales tâches auxquelles s'attaque la fouille de textes. Chacune de ces tâches sera un cas particulier du schéma général de la **figure 1 .6** pour lequel nous préciserons :

- la nature des données et des résultats (en particulier, s'il s'agit de textes, quelle représentation est privilégiée).
- la nature des ressources utiles, à titre obligatoire ou facultatif.
- la nature des méthodes utilisées pour la programmer, et si elle peut être abordée par apprentissage automatique.
- les applications concrètes de cette tâche.

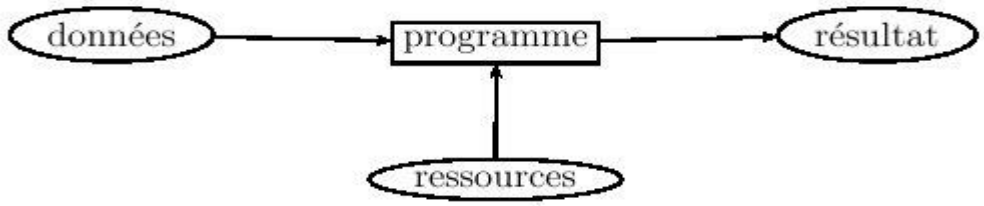


Figure1.6: Schéma général d'une tâche de la fouille de textes[9].

2-6-1 Tache de la texte fouille de text

A- Classification :

La tâche la plus "naturelle" à envisager, étant donnée la section précédente, est la classification de textes. Elle consiste à ranger des textes ou des documents dans des "classes" prédéfinies :

- **les données :** sont donc des textes, la plupart du temps représentés sous la forme de vecteurs. Des variantes de ce type de représentation ont été étudiées spécialement pour cette tâche, par exemple pour donner plus d'importance aux mots présents dans des titres, ou privilégier certaines catégories grammaticales.
- **les ressources** nécessaires sont celles qui permettent la représentation du texte : anti dictionnaire, lemmatiseur voire analyseur morphologique, compte d'occurrences, étiqueteur "part of speech" si on privilégie certaines catégories...
  Cette tâche est presqu'exclusivement abordée par apprentissage automatique, à partir d'exemples de textes déjà classés.
- De manière générale, la classification automatique de textes par "thème" peut rendre de grands services. On peut aussi utiliser des méthodes similaires pour retrouver l'auteur d'un texte (l'étiquette de la classe est alors un nom d'auteur)
- l'autre type d'application en plein développement de la classification est la reconnaissance automatique des opinions véhiculées par un texte : les classes, dans ce cas sont par exemple "favorable" et "défavorable". Certaines sociétés qui reçoivent des courriers électroniques de consommateurs à propos de leurs produits s'en servent pour analyser leur contenu. Dans ce cas, la représentation des textes a intérêt à privilégier les adjectifs et les verbes, qui sont les principaux moyens d'exprimer une opinion.

[9] https://www.lattice.cnrs.fr/sites/itellier/poly_info_ling/linguistique009.html#toc21

B- **La *recherche d'information* (ou RI)** est l'autre "tâche" générale d'ors et déjà omniprésente dans nos usages quotidiens des ordinateurs. Nous la sollicitons chaque fois que nous recherchons des documents répondant à une "requête".

- **la donnée** fournie par l'utilisateur est donc une requête. Celle-ci peut prendre des formes diverses, suivant le niveau d'expertise de cet utilisateur et la structure de la base de documents à interroger : simple liste de mots clés, langage de requête structuré (combinaisons de critères booléens, expressions rationnelles, requêtes type SQL...), voire document "exemple" dont on cherche des exemplaires "proches" parmi un ensemble de textes.
- **les ressources** sollicitées sont tout d'abord le corpus de textes ou de documents que l'on cherche à interroger. Ce peut être une base d'articles, une encyclopédie, ce peut être Internet... Comme précédemment, il est éventuellement fait appel aux ressources nécessaires à la représentation de la requête par un vecteur. Enfin, quand la requête est réduite à un ensemble de mots-clés.

  On distingue trois familles de méthodes pour aborder la RI :

  1) **les méthodes *booléennes*** fonctionnent à l'aide d'un simple index qui donne, pour chaque unité lexicale figurant dans la requête, la liste des textes où cette unité est présente. Les requêtes acceptées sont alors généralement des combinaisons de critères booléens (avec les opérateurs NON, ET, OU). Des calculs simples permettent d'obtenir la liste des textes où tous ces critères sont satisfaits en même temps.
  2) **les méthodes *vectorielles***, comme leur nom l'indique, codent toutes les informations (la requête et les documents de la base) sous la forme de vecteurs. La représentation TF-IDF est née dans ce contexte, et y est particulièrement efficace. La RI se ramène alors à trouver les vecteurs les plus "proches" d'une vecteur donné (celui représentant la requête). Pour quantifier ces distances, on utilise souvent des mesures basées sur le cosinus de l'angle qu'ils font entre eux (facile à calculer par des formules mathématiques).
  3) **les méthodes *statistiques*** qui en fait reviennent à faire de la classification automatique en supposant que l'on connaît déjà, pour la requête, un ensemble de documents "pertinents" et de documents "non pertinents", et que l'on cherche à trouver tous les documents devant être classés comme pertinents. On

le voit, cette méthode n'est pas vraiment comparables aux autres, puisqu'elle fait des hypothèses supplémentaires sur ce qui doit être fourni au système. Mais c'est la seule manière de faire intervenir de l'apprentissage automatique dans la tâche de recherche d'information.

2-6-2 Recherches sur Internet :

Est, bien sûr, l'application phare de cette tâche. Les moteurs de recherche mettent en œuvre des méthodes booléennes : leur index fait leur force ! Or ces méthodes ne permettent pas de classer en "plus ou moins pertinent" les documents obtenus (en l'occurrence les sites Web).

C) *l'extraction d'information* :

La dernière tâche fondamentale que nous voulons présenter ici. Comme son nom l'indique, elle se fixe comme objectif d'*extraire* de textes des informations factuelles précises.

- **les données** d'entrées sont des représentations de textes de même type, où la notion de *séquencea2* est préservée, elles peuvent aussi être des *documents structurés* (pages HTLM ou XML) ; les sorties sont des *données structurées*, en général sous la forme d'une liste d'attributs (prédéfinis) remplie ; parmi les informations disponibles au wrapper, on suppose qu'il y a la liste des champs à extraire (cette liste dépend bien sûr du type de textes).
- **Les ressources** linguistiques utiles à la réalisation de cette tâche dépendent de la méthode employée : toutes les techniques d'identification d'entités nommées (liste de valeurs possibles, mais aussi expressions régulières ou automates) sont intéressantes car, souvent, la plupart des données à extraire (noms propres ou valeurs numériques) sont des entités nommées. Des étiqueteurs grammaticaux, voire des analyseurs syntaxiques, sont parfois aussi employés[10].

3- Fouille d'itemsets fréquents

Les itemsets fréquents sont des motifs ou patterns (tel que les ensembles d'items, les sous séquences, ou les sous structures) qui apparaissent fréquemment dans un ensemble de données. Par exemple, un ensemble d'items tel que le lait et le pain qui apparaissent souvent

[10] https://www.lattice.cnrs.fr/sites/itellier/poly_info_ling/linguistique009.html#toc21

dans une base de transactions dans un supermarché, est un ensemble d'items fréquent. Une sous séquence telle que acheter premièrement un PC puis une caméra numérique ensuite une Carte mémoire qui se produit souvent dans la base historique des achats, est une séquence d'items fréquente.

Les sous structures peuvent être des sous-graphes ou des sous-arbres qui peuvent être combinés avec des ensembles ou des séquences d'items Trouver de tels itemsets fréquents joue un rôle essentiel dans la fouille des associations et des corrélations, et

représente une tâche importante en fouille de donnes et constitue toujours un thème qui attire beaucoup de recherches.

L'analyse des d'itemsets fréquents trouve son application dans plusieurs domaines :

- L'analyse du panier du marché, pour comprendre les habitudes des clients afin de Mieux organiser les rayons d'articles, organiser les promotions.
- L'analyse d'ADN en biologie afin de comprendre les propriétés génétiques des espèces.
- L'analyse du climat en météorologie afin de mieux orienter l'agriculture ou choisir L'orientation des pistes des aérodromes. (DJEFFAL, 2014)

3-1 Concepts de base

3-1-1 **Base de données formelle** *:*

La version de base de l'extraction d'itemsets fréquents permet de faire la fouille dans une Table d'une base de données relationnelle dont les valeurs sont des booléens indiquant la

Présence ou l'absence d'une propriété. Une telle base est appelée **base de données formelle.**

Une base de données formelle est définie par un triplet (O; P;R) où :

– O est un ensemble fini d'objets.

– P est un ensemble fini de propriétés.

– R est une relation sur O × P qui permet d'indiquer si un objet x a une propriété p (Noté xRp) ou non.

Par exemple dans le cas d'analyse du panier dans un supermarché, O est l'ensemble des Transactions d'achat, P est l'ensemble d'articles et R est la relation indiquant si un article a est acheté dans la transaction t.

Considérons par exemple la base de données formelle suivante :

Tableaux 1.1**:** Base de données formelle

R	A	B	C	D	E
X1	1	0	1	1	0
X2	0	1	1	0	1
X3	1	1	1	0	1
X4	0	1	0	0	1
X5	1	1	1	0	1
X6	0	1	1	0	1

O = {x1; x2; x3; x4; x5; x6}.

P = {A,B,C,D,E}.

xRp si et seulement si la ligne de x et la colonne de p se croisent sur un 1

(et pas sur un 0), par exemple :x1Ra; x1Rc et x1Rd.

3-2 Itemset

Un itemset d'une base de données formelle (O; P;R) est un sous-ensemble de P. l'ensemble de tous les motifs d'une base est donc l'ensemble des parties de P, noté 2^P . On

dira qu'un objet x ϵ O possède un itemset m si ∀ p ϵ m; xRp. Pour la base de données en

exemple, on a donc :

itemset de taille 0 = ∅; (C_5^0 = 0 itemset).

– itemset de taille 1 = {a}; {b}; {c}; {d} et {e}, qu'on notera, pour simplifier, a; b; c; d Et e. (C_5^1 = 5itemset).

– itemset de taille 2 = ab; ac; ad; ae; bc; bd; be; cd; ce; de (C_5^2 = 10itemset)

– itemset de taille 3 = abc; abd; abe; acd; ace; ade; bcd; bce; bde; cde (C_5^3 = 10itemset).

– itemset de taille 4 = abcd; abce; abde; acde; bcde (C_5^4 = 5itemset).

–itemset de taille 5 = abcde (C_5^5= 1itemset).

Dans la base formelle précédente, x1 possède les motifs Ø ; a; c; d; ac; ad; cd et acd. (DJEFFAL, 2014).

3-2-1 Taille d'un itemset

La taille t d'un itemset **mi** est définie comme le **cardinal de ses attributs**, c'est-à dire **t (mi) = | mi |**.

3-2-2 Couverture

La couverture d'un **itemset** est l'ensemble des exemples couverts d'itemsets .

3-2-3 Support d'un itemset

Le support d'un itemset est le **cardinal de la couverture** d'itemsets.

3-2 Fréquence d'un itemset

• La fréquence d'un itemset est égale à son **support divisée par le nombre total d'exemples.**[11]

3-2-1 itemset fréquent

Soit $\sigma s \in [0; 1]$. Un itemset m est fréquent (sous-entendu, relativement au seuil σs) si Support(m) $\geq \sigma s$. Sinon, il est dit non fréquent.

Type des itemsets fréquents

Selon la nature des itemsets fréquents on peut trouver deux types :

- **itemset fréquent fermé :**
  Un itemset fréquent est dit fermé s'il ne possède aucun sur-itemset qui a le même support.
- **itemset fréquent maximal :**
  Un itemset fréquent est dit Maximal si aucun de ses sur-itemset immédiats n'est fréquent.

[11] http://www.kdnuggets.com/.

Exemple :

Le schéma suivant illustre la relation entre les itemsets fréquents, fréquents fermés et fréquents Maximaux :

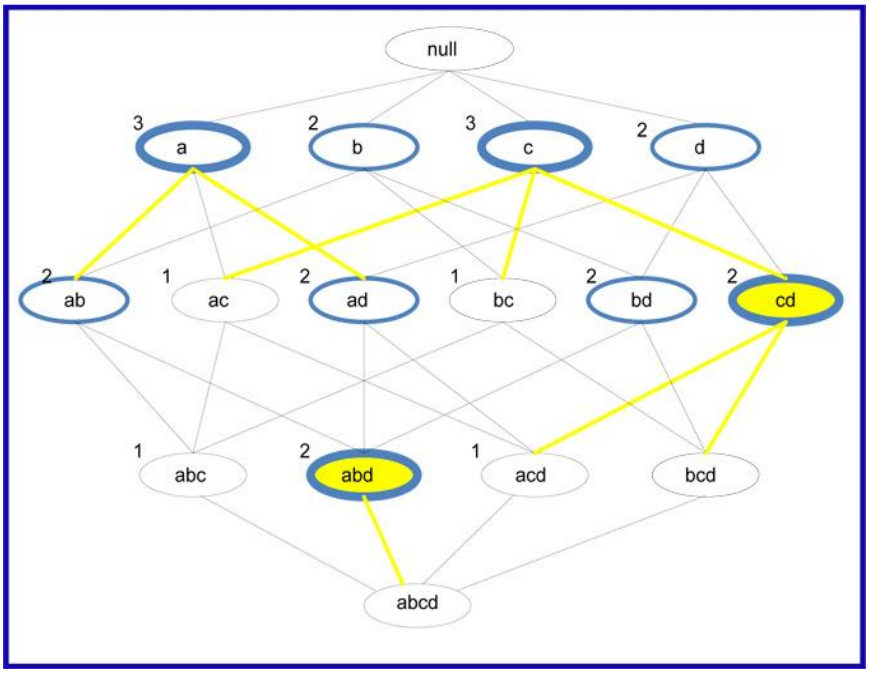


Figure 1.7 **:** La relation entre les itemsets fréquents, fréquents fermés et fréquents maximaux. (DJEFFAL, 2014) .

– Les itemsets encerclés par les lignes minces ne sont pas fréquents, les autres le sont.

– Les itemsets encerclés par des lignes plus épais sont fermés.

– Les itemsets colorés sont maximaux.

3-3-3 Passage aux règles d'association

La phase qui suit la phase de recherche des itemsets fréquents est la phase de découverte des règles d'association basées sur tous les items fréquents trouvés. Cette phase est relativement simple, elle consiste à trouver toutes les règles qui existent entre les items fréquents. Par exemple pour une règle telle que {x1; x2; x3}→x4, il faut premièrement que {x1; x2; x3} et x4 soient fréquents. Ensuite, il faut que la confidence de la règle dépasse un certain seuil. La confidence est calculée comme suit :

confidence ({x1; x2; x3}→x4) = P(x4 /({x1; x2; x3})

$$= \frac{support(\{x1,x2,X3\} \cup X4)}{SUPPORT(\{x1,x2,x3\})}$$

Où le Support(x) est le nombre d'enregistrements où apparait l'item x,

et le Support ({x1; x2; x3}) est le nombre d'enregistrement où apparaissent les itemset x1; x2; x3.

L'ensemble des règles d'association peut être trouvé en calculant la confidence de toutes Les combinaisons possibles des items fréquents puis prendre celles dont la confidence est importante. Cette opération peut être accélérée en enregistrant la liste des items fréquents

avec leurs fréquences dans une table qui peut être accédée rapidement.

3-3-4 Définition d'une règle d'association

Soit m1 et m2 deux itemsets . Une règle d'association est une implication de la forme :

$m1 \rightarrow m2$

Où m1, m2 $\in 2^p$; et $m1 \cap m2 = \emptyset$,

La règle $m1 \rightarrow m2$ est vérifiée dans la base de donnée D avec un support s, où s est le pourcentage d'objets dans D contenant $m1 \cup m2$:

Support $(m1 \rightarrow m2) = \frac{nombre\ de\ transaction\ contenant\ (m1 \cup m2)}{nombre\ totale\ de\ transaction}$

La confidence de la règle $m1 \rightarrow m2$ est définie par le pourcentage de transactions qui contiennent $m1 \cup m2$ dans les transactions contenant m1.

Confidence $(m1 \rightarrow m2) = \frac{nombre\ de\ transactin\ (m1 \cup m2)}{nombre\ de\ transactioncontenant\ m1}$

Les règles qui dépassent un minimum de support et un minimum de confidence sont Appelées règles **solide.**

Une fois les itemsets fréquents dans une base de données sont extraits, il devient simple de générer les règles d'association qui vérifient un minimum de support et un minimum de confidence, comme suit :

– Pour chaque itemset fréquent l, générer tous les sous ensembles non vides de l.

– Pour chaque sous-ensemble non vide s de l, enregistrer la règle $(s \rightarrow l\text{-}s)$ si :

$confidence(s \rightarrow l - s) \geq Min\_Conf$

où Min_Conf est un seuil minimum de confidence.

puisque les règles sont générées des itemsets fréquents, chacune vérifie automatiquement Le support minimum.

3-4 Méthodes efficaces pour la recherche des itemsets fréquents

Une approche naïve pour l'extraction des itemsets fréquents consiste à parcourir l'ensemble

de tous les itemsets, à calculer leurs nombres d'occurrences (support) et à ne garder que les plus fréquents. Malheureusement, cette approche est trop consommatrice en temps et en ressources. En effet, le nombre d'itemsets est 2^p (p est le nombre de propriétés), et en pratique, on veut manipuler des bases ayant un grand nombre d'attributs. L'algorithme apriori proposé par Agrawal et ses co-auteurs en 1994 est un algorithme de base qui permet d'extraire des itemsets fréquents dans une base ayant plusieurs milliers d'attributs et plusieurs millions d'enregistrements. L'idée est d'effectuer une extraction par niveaux selon le principe suivant :

– On commence par chercher les itemsets fréquents de longueur 1 ;

– On combine ces itemsets pour obtenir des itemsets de longueur 2 et on ne garde que les fréquents parmi eux ;

– On combine ces itemsets pour obtenir des itemsets de longueur 3 et on ne garde que les fréquents parmi eux ;

– continuer jusqu'à la longueur maximale.

cette approche s'appuie sur les deux principes fondamentaux suivants (qui reposent sur la décroissance du support) :

1. Tout sous-itemset d'un itemset fréquent est fréquent.
2. Tout sur-itemset d'un itemset non fréquent est non fréquent.

L'algorithme de référence basé sur cette approche est l'algorithme Apriori. Le pseudo code suivant décrit l'extraction d'itemsets fréquents selon ce principe :

Algorithme 1 Apriori

Require: Base de données de transactions D, Seuil de support minimum σ

Ensure: Ensemble des items fréquents

i ← 1

C1 ← ensemble des items de taille 1 (un seul item)

While Ci≠ ∅ **do**

Calculer le Support de chaque itemset m ∈ Ci dans la base

Fi ← {m ∈ Ci / support(m)≥ σ}

Ci+1 ← toutes les combinaisons possibles des itemsets de Fi de taille i + 1

i ← i + 1;

end while

retourner ($U_{i\geq 1}$) Fi

Exemple :

L'application de l'algorithme sur la base donnée en exemple avec σ = 0,25

se passe comme suit :

1. **Génération de candidats de taille 1** :

– c1 = {a; b; c; d; e}

– Supports : $\frac{3}{6}, \frac{5}{6}, \frac{5}{6}, \frac{1}{6}, \frac{5}{6}$

D'où F1 = {a; b; c; e} (aucun items fréquent ne contiendra d).

2. **Génération de candidats de taille 2** : Combiner 2 à 2 les candidats de taille 1 de F1 :

– C2 = {ab; ac; ae; bc; be; ce}

– Supports : $\frac{2}{6}, \frac{3}{6}, \frac{2}{6}, \frac{4}{6}, \frac{5}{6}, \frac{4}{6}$

F2 = C2 : tous les itemsets de C2 sont fréquents.

3. Génération de candidats de taille 3 : Combiner 2 à 2 les candidats de taille 2 de F2

(Et ne considérer que ceux qui donnent des itemsets de taille 3) :

– C3 = abc; abe; ace; bce

– Supports : $\frac{2}{6}, \frac{2}{6}, \frac{2}{6}, \frac{4}{6}$

F3 = C3 : tous les itemsets de C3 sont fréquents.

4. Génération de candidats de taille 4 :

– C4 = {abce}

– Supports : $\frac{2}{6}$,

5. Génération de candidats de taille 5 : C5 =∅ ;. Donc, F5 = ∅;

6. L'algorithme retourne alors l'ensemble des itemset fréquents : F1 ∪F2 ∪ F3 ∪ F4

Remarque 1 : On peut voir cet algorithme comme le parcours du treillis des parties

de P ordonné pour l'inclusion.

supposons par exemple que P = {a; b; c}. Le treillis des parties de P est :

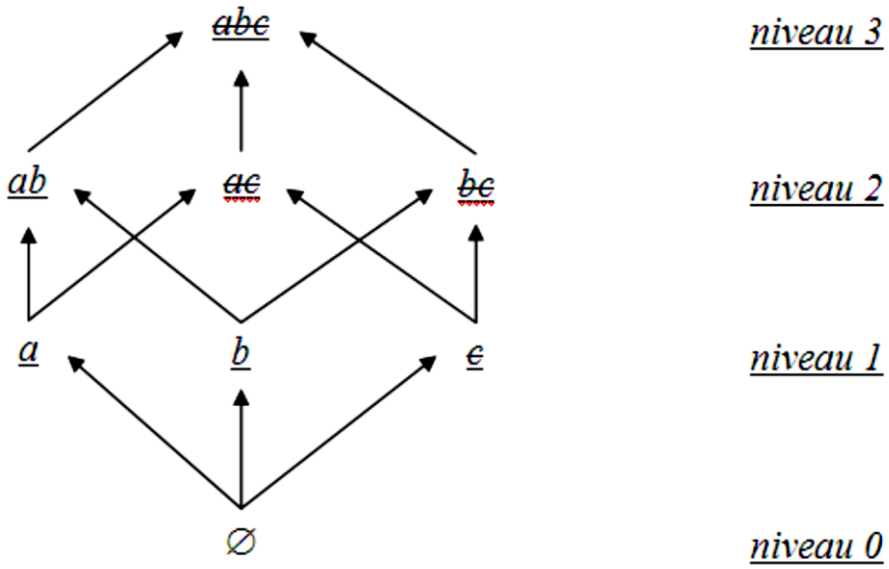


Figure 1.8 : Le Treillis Des Parties De P (DJEFFAL, 2014)

Il est parcouru par niveau croissant à partir du niveau i = 1. Quand un itemset n'est pas fréquent, tous ses sur-items sont non fréquents. Dans notre exemple c n'est pas fréquent (il a été barré) et, par conséquent, aucun de ses sur-items n'est considéré. On a ainsi élagué le parcours du treillis.

Remarque 2 : Un des objectifs des optimisations de cet algorithme est de diminuer Le nombre d'accès à la base de données.

Remarque 3 : Le seuil σ est fixé par l'analyste. Celui-ci peut suivre une approche Itérative en fixant un seuil au départ et, en fonction du résultat, changera la valeur du Seuil : Si trop de itemset fréquents ont été trouvés, il augmentera le seuil ; dans le cas inverse, Il le diminuera. Par ailleurs, on peut constater que le temps de calcul de l'algorithme Apriori Décroit avec le seuil. Par conséquent, si l'analyste fixe une valeur de seuil trop grande, cela Gaspillera moins de temps que s'il en fixe un trop petit**.**

4- ensemble d'éléments à haute utilité (High Utility Itemsets HUI)

Un ensemble d'éléments à haute utilité est un ensemble de valeurs qui apparaît dans une base de données et qui a une grande importance pour l'utilisateur, telle que mesurée par une fonction d'utilité. L'extraction d'ensembles d'éléments à haute utilité généralise le problème de l'extraction fréquente d'éléments en considérant les quantités et les poids des éléments.[12]

12 https://www.google.com/search?client=opera&q=4.%09High+Utility+Itemsets+HUI&sourceid=opera&ie=UTF-8&oe=UTF-8

4-1 Frequent itemset mining

Le problème de l'extraction d'itemsets à haute utilité est une extension du problème de l'extraction fréquente de motifs. Le minage de motifs fréquents est un problème populaire en fouille de données, qui consiste à trouver des motifs fréquents dans les bases de données de transactions.

4-2 Exploitation d'ensembles d'éléments à haute utilité (HUIM)

Pour remédier à ces limitations, le problème de l'extraction fréquente d'itemsets a été redéfini comme le problème de l'extraction d'itemsets à haute utilité. Dans ce problème, une base de données de transactions contient des transactions où les quantités d'achat sont prises en compte ainsi que le profit unitaire de chaque article. Par exemple, considérez la base de données de transactions suivante :

Tableaux 1.2 **:** Base de données de transactions avec des euantités et des Informations. (Philippe, 2015)

Transaction database with quantities

Transaction.	Items
T0	a(1), b(5), c(1), d(3), (e, 1)
T1	b(4), c(3), d(3), e(1)
T2	a(1), c(1), d(1)
T3	a(2), c(6), e(2)
T4	b(2), c(2), e(1)

unit profit table

Item	Unit profit
A	5$
B	2$
C	1$
D	2$
E	3$

Considérez la transaction T3. Il indique que le client correspondant a acheté deux unités de l'article "a", six unités de l'article "c" et deux unités de l'article "e". Regardez maintenant le tableau de droite. Ce tableau indique le profit unitaire de chaque article. Par exemple, le profit unitaire des éléments « a », « b », « c », « d » et « e » est respectivement de 5$, 2$, 1$, 2$ et 3$. Cela signifie par exemple que chaque unité de "a" qui est vendue génère un profit de 5$.

Le problème de l'exploration d'ensembles d'éléments à haute utilité est de trouver les ensembles d'éléments (groupe d'éléments) qui génèrent un profit élevé dans une base de données lorsqu'ils sont vendus ensemble. L'utilisateur doit fournir une valeur pour un seuil appelé "minutil" (le seuil d'utilité minimum). Un algorithme d'extraction d'itemsets à haute utilité génère tous les itemsets à haute utilité, c'est-à-dire ceux qui génèrent au moins un profit « minutil ». Par exemple, considérons que "minutil" est défini sur 25 $ par l'utilisateur. Le résultat d'un algorithme d'exploration d'ensembles d'éléments à haute utilité serait le suivant :

Tableaux1.3: **Ensemble d'éléments à haute utilité High-Utility Itemsets (Philippe, 2015)**.

Ensemble d'éléments à haute utilité	
{a,c}:28$	{a,c,e}:31$
{a,b,c,d,e}:25$	{b,c}:28$
{b,c,d}:34$	{b,c,d,e}:40$
{b,c,e}:37$	{ b,d}:30$
{b,d,e}:36$	{b,e}:31$
{c, e}:27$	

4-3 problème de l'exploration

- Premièrement, il peut être plus intéressant d'un point de vue pratique de découvrir des ensembles d'articles qui génèrent un profit élevé dans les transactions des clients que ceux qui sont achetés fréquemment.
- Deuxièmement, du point de vue de la recherche, le problème de l'exploration d'ensembles d'éléments à haute utilité est plus difficile. Dans l'exploration fréquente d'itemsets, il existe une propriété bien connue de la fréquence (support) des itemsets qui stipule qu'étant donné un itemset, tous ses sur-ensembles doivent avoir un support inférieur ou égal. Ceci est souvent appelé la "propriété Apriori" ou la propriété "anti-monotonicité" et est très puissant pour élaguer l'espace de recherche car si un ensemble d'éléments est peu fréquent, nous savons que tous ses sur ensembles sont également peu fréquents et peuvent être élagués. Dans l'extraction d'itemsets à haute utilité, une telle propriété n'existe pas.

5- Conclusion

Nous avons présenté dans la premier section de chapitre les deux technologies fouille de donnes et plus en détail le fouille de texte, qui divisé en deux étapes Principales, étape d'analyse qui permet de structurer le texte, et une étape d'interprétation de l'analyse, qui fait appel aux méthodes de fouille de données, en plus nous avons présenté la fouille des motif fréquents, Le chapitre suivant présente une description détaillée du traitement des langages naturel et la (WSD).

Chapitre 02 : Desambiguation du sens du mots DSM

Chapitre2 Desambiguation du sens du mots

Introduction

Le Traitement Automatique Du Langage Naturel (TALN) est un domaine de recherche multidisciplinaire, dont l'objectif vital est de développer des théories, des algorithmes et des innovations qui permettent et renforcent la communication entre les ordinateurs et les humains en utilisant des langues qui ont naturellement évolué dans les sociétés humaines (standard exemple, l'anglais, espagnol, français, entre autres.) au lieu des langages formels construits qui ont été employés pour software engainer les ordinateurs. Les exemples d'applications de la TALN incluent la gestion et la découverte des connaissances, la recherche d'informations, la réponse aux questions et la traduction automatique. L'un des grands déterrent à l'interaction homme-machine est la prévalence des homonymes dans de nombreuses langues naturelles (c'est-à-dire des mots qui sont prononcés ou épelés de la même manière mais qui ont des implications différentes). En général, les humains sont très doués pour comprendre le sens des mots ambigus ; cependant, la désambiguïsation automatique des mots reste une tâche difficile pour les ordinateurs.

La désambiguïsation du sens des mots (DSM) est l'un des sujets centraux de la TAL. WSD consiste à trouver automatiquement le sens correct d'un mot ambigu dans un texte, simplement en analysant le contexte dans lequel il existe. Les méthodes WSD actuelles peuvent être classées en quatre catégories: supervisées, non supervisées, semi-supervisées et basées sur les connaissances.

1- Traitement Automatique du Langage Naturel (TLN ou NLP)

1-1 Définition

Le Traitement Du Langage Naturel (TLN, ou NLP en anglais) est la capacité pour un programme informatique de comprendre le langage humain tel qu'il est parlé. Il fait partie des technologies d'intelligence artificielle.

Le développement d'applications TLN est difficile parce que les ordinateurs sont conçus pour que les humains leur « parlent » dans un langage de programmation précis.[13]

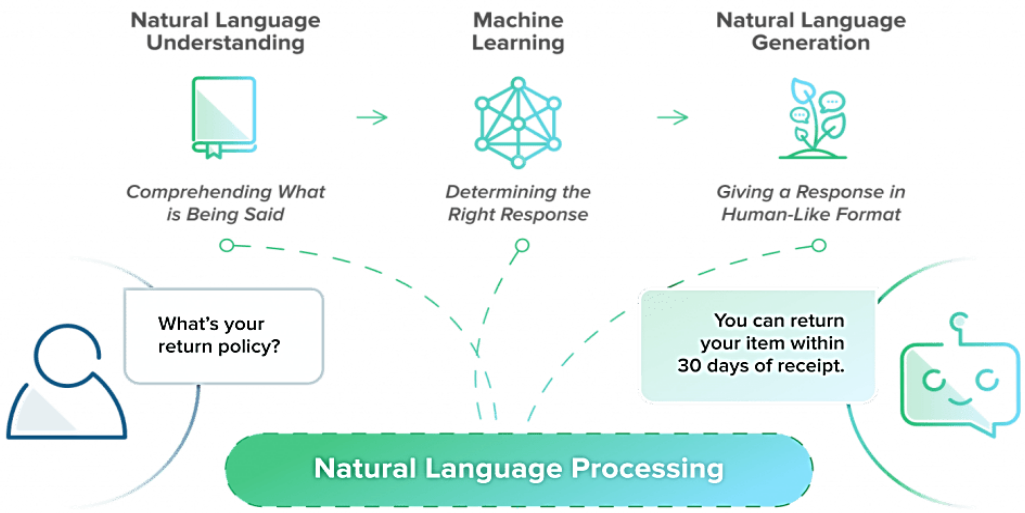


Figure 2.1 : Traitement de Langage Natural. (Mansour, 2021).

1-2 Classification du traitement du langage naturel :

Traitement du langage naturel. Il se regroupe en deux domaines spécifiques (Figure2.2) :

[13] https://www.lemagit.fr/definition/Traitement-du-langage-naturel-TLN

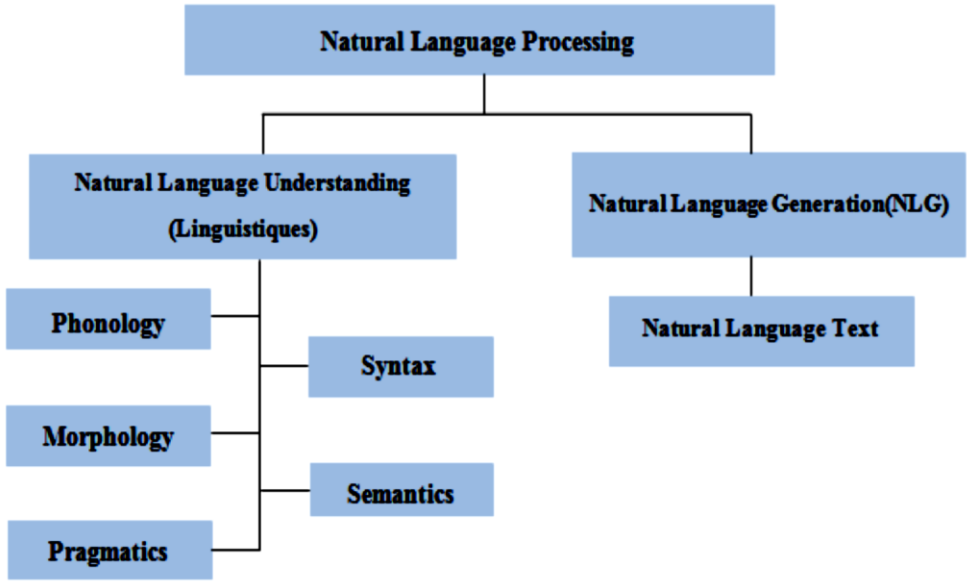


Figure 2.2 : Classification générale du traitement du langage naturel (Diksha Khurana, 2017).

1-2-1 Compréhension du langage naturel (Natural Language Understanding (NLU)):

Est une sous-rubrique du traitement de la langue naturelle en intelligence artificielle qui traite de la compréhension en lecture automatique. La compréhension du langage naturel est considérée comme un problème difficile en IA. Il tente de résoudre l'ambiguïté suivante en langage naturel.

- Ambiguïté lexicale - les mots ont plusieurs sens;
- Ambiguïté syntaxique - State comprenant plusieurs arbres d'analyse ;
- Ambiguïté sémantique - State à sens multiples. (Curry, 2002)

1-2-2 Génération de langage naturel (Natural Language Génération (NLG))

Quand nous parlons de génération de langage naturel, nous nous référons à une technologie qui est une synthèse entre l'informatique, l'intelligence artificielle et la linguistique.

Il s'agit en fait d'un logiciel qui recueille une grande quantité de données, les traite et, enfin, parvient à reproduire le langage humain.

Cette technologie ne répond pas à des réponses prédéterminées, mais est capable de s'adapter au contexte et aux interactions linguistiques, elle est capable de réellement comprendre le texte dans toutes ses implications sémantiques et signifiantes.

Génération du langage naturel divisée en trois étapes proposées :

- Planification de texte - l'ordre de base du contenu dans les données structurées est Effectué.
- Planification des peines - les expressions sont combinées à partir de données structurées pour représentent le flux d'informations.

Réalisation - des expressions grammaticalement correctes sont finalement produites pour représenter le texte. (Cannobio, 2021)

2.2.3 Les méthodes principales utilisées en TAL

Généralement il y a deux aspects essentiels pour tout problème de TAL:

- ✓ ***La partie « linguistique » :***
  Qui permet de prétraiter et transformation les informations en un jeu de données exploitable.
- ✓ *La* ***partie « apprentissage automatique :***
  C'est l'application de méthode de machine Learning ou deep learning à ces données.

En explique brièvement les principales méthodes de NLP et en mettant en avant les principaux challenges.

Nous avons choisi un exemple classique : **la détection des spams .**

1-2-4 La phase de prétraitement du texte aux données

On va déterminer que si un mail est un spam ou no a partir de son contenu

Pour ce là on va transformer des donnée brutes en des donnée exploitable.

Parmi les principales étapes, on retrouve :

- **Nettoyage :** cette phase consiste à réaliser des tâches telles que la suppression d'urls, d'emoji ……etc.
- **Normalisation des données :**
  - ✓ **Tokenisation :** ou découpage du texte en plusieurs pièces appelés *tokens*.
    Exemple : « Vous trouverez en pièce jointe le document en question »

; « Vous », « trouverez », « en pièce jointe », « le document », « en question ».

- ✓ **Stemming :** Le **stemming** désigne généralement le processus heuristique brut qui consiste à découper la fin des mots dans afin de ne conserver que la racine du mot. *Exemple : « trouverez »* -> *« trouv »*
- ✓ **Lemmatisation :** permet de faire la réalisation de même tache mais en utilisant une analyse fine de la construction des mots, donc en supprime uniquement les terminaisons inflexibles (isoler la forme canonique du mot).
- ✓ **Autres opérations :** suppression des chiffres, ponctuation, symboles et *stopwords*, passage en minuscule.

Afin de pouvoir appliquer les **méthodes de Machine Learning** aux problèmes relatifs au langage naturel, il est indispensable de transformer les données textuelles en données numériques.

Principalement il existe plusieurs approches sont les suivants :

- ✓ **Term-Frequency (TF) :** cette méthode consiste à compter le nombre d'occurrences des *tokens* présents dans le corpus pour chaque texte.
  On représente chaque texte par un vecteur d'occurrences**. (bag_of_word).**

Tableau 2.1 : Représentation des vecteurs issues de la méthode term-frequency (TF) (lina, 2020).

	Doc1	**doc2**	**doc3**	**doc4**	**doc5**	**doc6**	**doc8**	**doc9**	
Term(s)1	10	0	1	0	0	0	0	2	
Term(s)2	0	2	0	0	0	18	0	2	
Term(s)3	0	0	0	0	0	0	0	2	
Term(s)4	6	0	0	4	6	0	0	0	~~word vector~~
Term(s)5	0	0	0	0	0	0	0	2	(passage vector)
Term(s)6	0	0	1	0	0	1	0	0	
Term(s)7	0	1	8	0	0	0	0	0	
Term(s)8	0	0	0	0	0	3	0	0	

Document vector

✓ **Term Frequency-Inverse Document Frequency (TF-IDF*):***

Cette méthode consiste à compter le nombre d'occurrences des *tokens* présents dans le corpus pour chaque texte, que l'on divise ensuite par le nombre d'occurrences total de ces même *tokens* dans tout le corpus. Pour le terme x présent dans le document y, on peut définir son poids par la relation suivante :

$$w_{x,y} = tf_{x,y} \times \log\left(\frac{N}{dfx}\right)$$

Où :

- $tf_{x,y}$ est la fréquence du terme x dans y ;
- df_x est le nombre de documents contenant x ;
- N est le total de document.

Le résultat de cette approche permet de donner une représentation vectorielle qui comporte des vecteurs de poids et non plus d'occurrences.

1-3-5 La phase d'apprentissage des données au modèle

On peut distinguer 3 principales approches TAL *:*

- L es méthodes basées sur des règles
- modèles classiques de Machine Learning
- modèles de Deep Learning.

➢ **Méthodes basées sur des règles :**

Les méthodes fondées sur des règles reposent en grande partie sur l'élaboration de règles spécifiques à un domaine (par exemple, les expressions régulières). Elles peuvent être utilisées pour résoudre des problèmes simples tels que l'extraction de données structurées à partir de données non structurées (par exemple, les pages web).

Dans le cas de la détection de spams, cela pourrait consister à considérer comme e-mails indésirables, ceux qui comportent des buzz words tels que « promotion », « offre limitée ».

- **Modèles classiques de Machine Learning :**

Les approches classiques d'apprentissage automatique peuvent être utilisées pour résoudre des problèmes plus difficiles. Contrairement aux méthodes fondées sur des règles prédéfinies, elles reposent sur des méthodes qui portent réellement sur la compréhension du langage.

- **Modèles de Deep Learning :**

L'utilisation de modèles profond pour les problématiques TAL fait l'objet de nombreuses recherches actuellement.

Ces modèles se généralisent encore mieux que les approches classiques d'apprentissage car ils nécessitent une phase de prétraitement du texte moins sophistiquée**.** (lina, 2020) **.**

2- Domaines d'application du NLP

Le NLP est une discipline recouvrant un champ d'application très large.

Voici les applications les plus populaires :

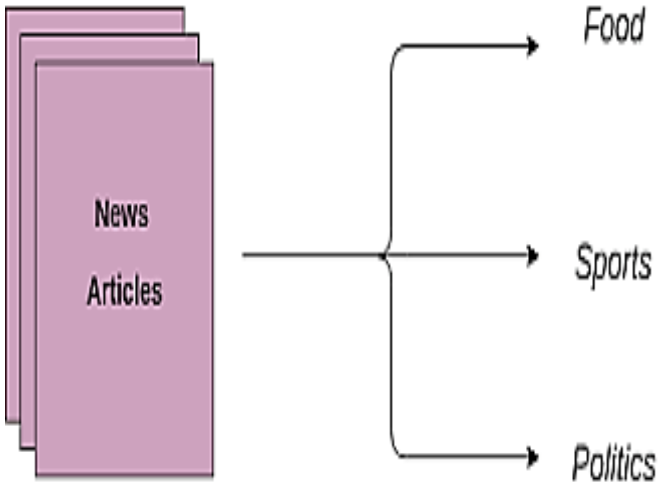


Figure 2.3 : Classification et catégorisation te Texte**.**

Il existe bon nombre d'applications du NLP comme la recherche Web.

La classification de textes consiste à attribuer une catégorie prédéfinie à un texte donné.

-Les agents conversationnels "Les Chatbots " :

Diverses techniques et méthodes de NLP sont au cœur du fonctionnement des Chatbots actuels. Les Chatbots sont capables de gérer des tâches standard telles que : renseigner les clients sur des produits ou services, les aiguiller, etc.

L'implémentation des agents conversationnels “Chatbots” s'appuient sur une sous-rubrique du NLP qu'on l’appelle NLU.

-Génération automatique de textes :

Cette technique est très utilisée dans le domaine de la santé, pour établir automatiquement un compte-rendu d’un patient suivi à l’hôpital à partir de son historique de comptes rendus médicaux.

-Reconnaissance de caractères :

Est la technique qui consiste à extraire le contenu textuel d’un document ainsi que des informations importantes à partir des documents non structurés, des factures.

-Résumé automatique :

Cette technique de TAL permet de produire des résumés courts et précis d’un document textuel plus long. (AMIA, 2022)

2-1 Le problème central du NLP l’ambiguïté

Parmi les problèmes les plus importants du traitement des langues naturelles, on trouve en premier rang le problème de l’ambiguïté (on dit qu'il y a ambiguïté lorsqu'il y a plus d'une interprétation pour une structure linguistique donnée). En effet, l'être humain parlant d’une langue donnée, passe à côté de la plupart des ambiguïtés, puisqu'il est très doué en la manière de les résoudre : il utilise ses Connaissances qu'il a accumulées à propos du monde et du contexte. Mais, malheureusement, les machines ne

sont pas encore aussi compétentes. Les applications du TALN sont souvent confrontées aux ambiguïtés à tous les niveaux d'analyse.

Exemple : « Le secrétaire vole des livres ».

Cette phrase comporte cinq mots ambigus, comme le montre le tableau suivant **:**

Tableaux 2.2 : Les cinq mots ambigus dans la phrase

N°	Mot	Sens
01	**Le**	Article ou pronom
02	**Secrétaire**	Homme, meuble, oiseau
03	**Voler**	Planer, dérober
04	**Des**	Article contracté ou article partitif
05	**Livre**	Bouquin, monnaie, poids

Elle correspond donc à plusieur possibilités. Le programme d'analyse devra lever ces ambiguïtés de diverses façons complémentaires, en considérant des règles morphologiques, syntaxiques, Sémantiques[14]

2-2 Désambiguïsation du sens des mots (WSD)

2-2-1 À propos de la désambiguïsation du sens des mots

La désambiguïsation nécessite deux entrées strictes **:**

- Un dictionnaire pour spécifier les sens qui doivent être désambiguïsés.
- un corpus de données linguistiques à désambiguïser (dans certaines méthodes, un corpus d'apprentissage d'exemples de langage est également requis).

La tâche WSD a deux variantes :

- échantillon lexical : désambiguïsation des occurrences d'un petit échantillon de mots cibles préalablement sélectionnés.
- tous les mots : désambiguïsation de tous les mots dans un texte courant. La tâche « Tous les mots » est généralement considérés comme une forme d'évaluation plus réaliste.

[14] https://www.loukam.net/TALN_Chap1.pdf.

2-2-2 implémenter WSD

Il existe quatre façons principales d'implémenter WSD :

- Méthodes basées sur le dictionnaire et les connaissances

  Ces méthodes reposent sur des données textuelles telles que des dictionnaires, des thésaurus, etc. Elles sont basées sur le fait que les mots qui sont liés les uns aux autres peuvent être trouvés dans les définitions.

- Méthodes supervisées

  Dans ce type, des corpus annotés en sens sont utilisés pour former des modèles d'apprentissage automatique. Mais, un problème qui peut

  Se pose est que de tels corpus sont très difficiles et longs à créer.

- Méthodes semi-supervisées

  En raison de l'absence d'un tel corpus, la plupart des algorithmes de désambiguïsation du sens des mots utilisent des algorithmes semi-supervisés.

  Méthodes. Le processus commence avec une petite quantité de données, qui est souvent créée manuellement. Ceci est utilisé pour former un classificateur initial. Ce classificateur est utilisé sur une partie non taguée du corpus, pour créer un plus grand ensemble de formation. Fondamentalement, cette méthode consiste à démarrer à partir des données initiales, qui sont appelées comme données de départ.

- Méthodes non supervisées

Les méthodes non supervisées posent le plus grand défi aux chercheurs et aux professionnels de la TAL. Une clé L'hypothèse de ces modèles est que des significations et des sens similaires se produisent dans un contexte similaire. Ils ne sont pas dépendant des efforts manuels, peut donc surmonter l'impasse de l'acquisition des connaissances. (Majumder, 2021)

2-3 Travaux connexes

De nombreux efforts ont été déployés pour résoudre le problème du WSD depuis son apparition. La recherche sur le WSD s'accélère en raison de ses nombreuses applications, En

conséquence, un certain nombre de nouveaux systèmes ont été développés et validés sur des standards ensembles de données qui sont spécialisés dans l'évaluation WSD.

En termes de performances, les approches WSD supervisées sont supérieures

dans la plupart des cas puisqu'ils peuvent apprendre le mappage entre

certaines caractéristiques et le sens de la cible à partir d'un relativement grand annoté

corpus. Ceci a également été vérifié par les résultats de plusieurs compétitions WSD. Les premières tentatives dans cette catégorie comprennent les arbres de décision.

IMS : est le premier système complet accessible au public pour WSD. En raison des grands progrès des techniques d'apprentissage en profondeur, Depuis quelques années, des systèmes plus supervisés sont proposés pour Faire face aux tâches WSD. **IMS** (**It makes sense**) est un système qui utilise pour désambiguïser les mots dans leur contexte.

une version étendue d'IMS a été développée, qui intègre les incorporations de mots comme fonctionnalité pour former les classificateurs (experts en mots) et améliore encore les performances.

- **SUPWSD :** est un autre cadre supervisé WSD nouvellement disponible. L'algorithme est le même que celui d'IMS mais il utilise beaucoup plus grand corpus de formation annoté en sens et offre plus de flexibilité pour la personnalisation. Plus important encore, le système **SUPWSD** fonctionne beaucoup mieux et plus rapidement qu'IMS. De plus, certaines profondes modèles de réseaux de neurones tels que bidirectionnel long court terme.
- **Lesk :** Lesk Algorithm est un algorithme classique de désambiguïsation du sens des mots introduit par Michael E. Lesk en 1986.
  L'algorithme de Lesk est basé sur l'idée que les mots d'une région donnée du texte auront une signification similaire. Dans l'algorithme de Lesk simplifié, la signification correcte de chaque contexte de mot est trouvée en obtenant le sens qui chevauche le plus le contexte donné et sa signification dans le dictionnaire.

3 Conclusion

Dans chapitre, nous avons passé en revue la tâche de désambiguïsation du sens des mots, DSM est les plus difficiles problèmes ouverts de la TAL. Certains des défis du DSM ont été

discutés, de nombreux problèmes surviennent en DSM parce qu'elle dépend de connaissances tirées de différentes ressources. DSM traite également avec la complexité des langues. Approche basée sur les connaissances, supervisées, non supervisée …….ets

Chapitre 3 :

Extraction d'éléments à haute utilité

(HUIM)

Chapitre 3 Extraction d'éléments à haute utilité (HUIM)

Introduction

Dans ce chapitre, nous présenterons la méthode que nous avons proposé pour la désambiguïsation du sens des mots (WSD), ainsi que les outils utilisés pour le développement du system tels que le choix du langage, l'environnement, ainsi que l'ensemble des résultats des expérimentations obtenus. Nous terminerons par une conclusion.

1- Vue globale du système proposé

Le but essentiel de notre travail est de déterminer le sens le plus approprié pour chaque mot cible d'un texte. Notre méthode consiste à estimer la proximité sémantique qui existe entre plusieurs sens de mot. L'idée centrale de ce système consiste à considérer la similitude entre le contexte d'un mot à désambigüiser et le contexte pertinent des informations telles que la définition d'un sens potentiel et son sens connexe obtenu.

Une vue globale de l'approche proposée est donnée par la figure (3.1)

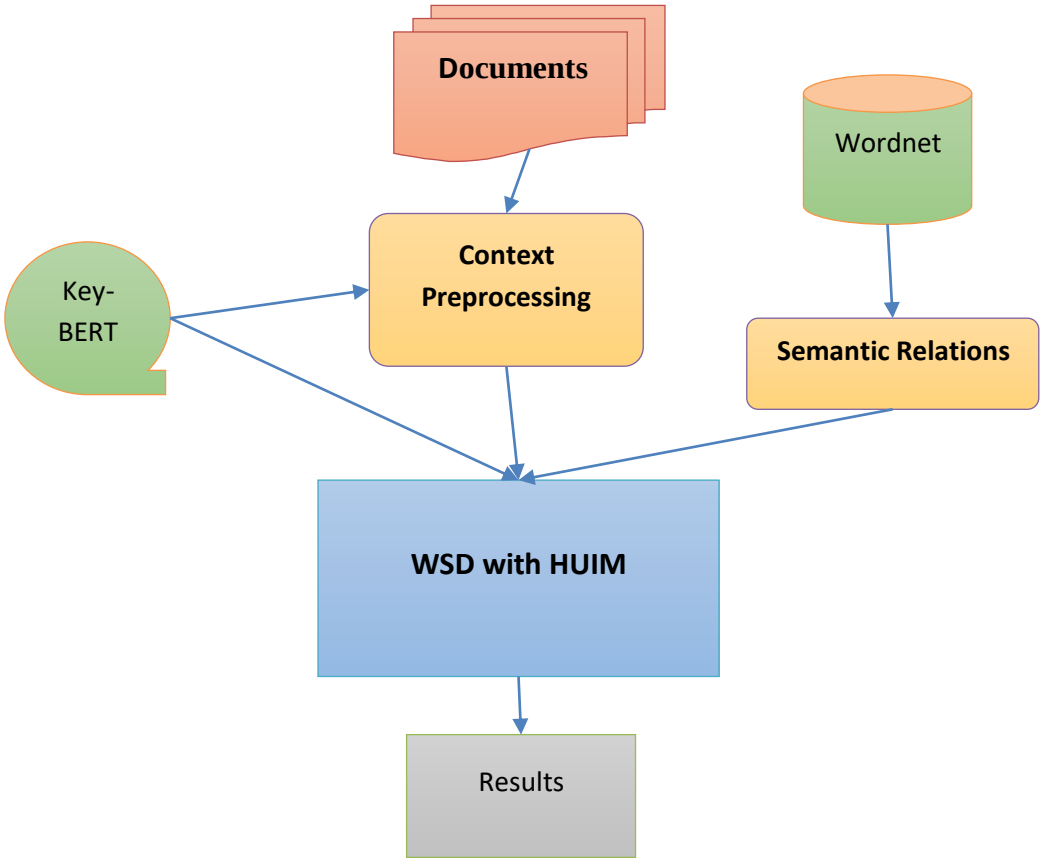


Figure 3.1 **:** Une vue globale de l'approche proposée.

Notre système prends comme entree n'importe quel type de documents (libre, semi-structuré). Ensuite, une étape d'extraction des mots-clés associée à chaque contexte d'un mot cible est appliquée en utilisant la ressource Key-BERT. En se basant sur les ressources Wordnet et Key-BERT, une liste des transactions est créée pour chaque sens du mot cible où chaque transaction est constituée d'une suite de mots associés à leur poids. Cette liste sera utilisée pour l'extraction des itemsets fréquents à haute utilité (HUI). La derniere étape consite à utiliser l'ensemble des HUI obtenus pour déterminer le sens approprié pour chaque mot cible.

2- Outil on corpus

1-SemEval (International Workshop on Semantic Evaluation):est une série d'ateliers de recherche internationaux sur le traitement du langage naturel (TAL) dont la mission est de faire progresser l'état actuel de l'art en analyse sémantique et d'aider à créer des ensembles de données annotées de haute qualité dans une gamme de problèmes de plus en plus difficiles en

sémantique du langage naturel. L'atelier de chaque année présente une collection de tâches partagées dans lesquelles les systèmes d'analyse sémantique computationnelle conçus par différentes équipes est présentée et comparés.[15]

SemEval est le lieu principal de la communauté TAL pour la proposition de nouveaux défis et pour l'évaluation empirique systématique des systèmes TAL. (Oskar Wysocki, 2020).

Figure 3.2 **:** Partie d'un document XML du corpus semeval.

2- KeyBERT

KeyBERT Est une technique d'extraction de mots clés minimale et facile à utiliser qui exploite les intégrations BERT pour créer des mots clés et des phrases clés qui ressemblent le plus à un document.[16]

- **Comment fonctionne KeyBERT ?**

L'extraction de mots-clés se fait en trouvant les sous-phrases dans un document qui sont les plus similaires au document lui-même. Tout d'abord, les incorporations de documents sont extraites avec BERT pour obtenir une représentation au niveau du document.[17]

- **Comment KeyBERT extrait-il les mots-clés ?**

KeyBERT extrait les mots clés en procédant comme suit :

[15] https://pages.github.com/
[16] https://maartengr.github.io/KeyBERT/
[17] https://www.google.com/search?client=opera&q=what+is+KeyBERT&sourceid=opera&ie=UTF-8&oe=UTF-8

- Le document d'entrée est intégré à l'aide d'un modèle BERT pré-formé. Vous pouvez choisir n'importe quel modèle BERT de votre choix parmi transformers. Cela transforme un morceau de texte en un vecteur de taille fixe qui représente l'aspect sémantique du document.
- Les mots-clés et les expressions (n-grammes) sont extraits du même document à l'aide de techniques de sac de mots. Il s'agit d'une étape classique que vous connaissez peut-être si vous avez déjà effectué une extraction de mots clés.

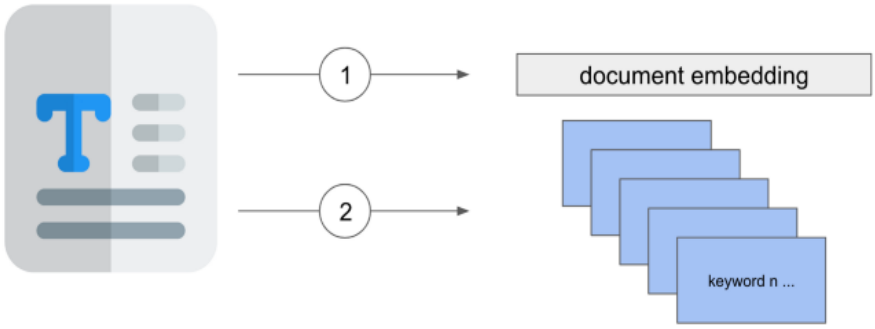


Figure 3.3 : Extraction des cots-clés. (Besbes, 2021)

- Chaque mot-clé est ensuite intégré dans un vecteur de taille fixe avec le même modèle utilisé pour intégrer le document.

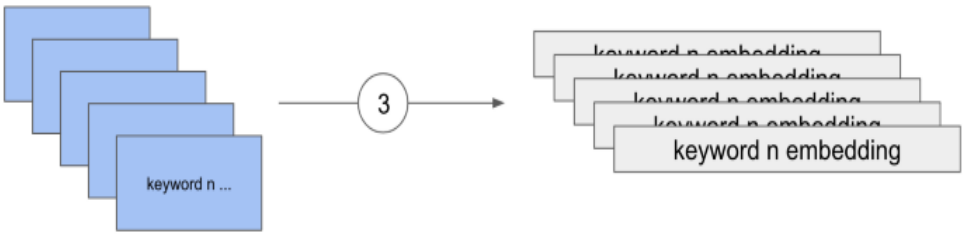


Figure 3.4 : Intégration des mots-clés dans des vecteurs. (Besbes, 2021)

- Maintenant que les mots-clés et le document sont représentés dans le même espace, KeyBERT calcule une similarité en cosinus entre les intégrations de mots-clés et l'intégration de document. Ensuite, les mots clés les plus similaires (avec le score de similarité cosinus le plus élevé) sont extraits. (Besbes, 2021)

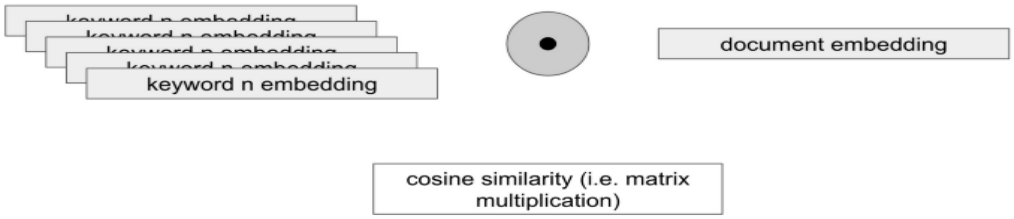


Figure 3.5 : Calcule des similarités. (Besbes, 2021)

3- WordNet

WordNet est une base de données lexicale de relations sémantiques entre des mots dans plus de 200 langues WordNet relie les mots en relations sémantiques, y compris les synonymes , les hyponymes et les méronymes . Les synonymes sont regroupés en synsets avec des définitions courtes et des exemples d'utilisation. WordNet peut donc être vu comme une combinaison et une extension d'un dictionnaire et d'un thésaurus. Bien qu'il soit accessible aux utilisateurs humains via un navigateur Web, son utilisation principale est en automatique analyse et applications d'intelligence artificielle. WordNet a été créé pour la première fois en anglais et la base de données et les outils logiciels en anglais de WordNet ont été publiés sous une licence de style BSD et sont disponibles gratuitement pour téléchargement à partir de ce site Web WordNet.

3-1 Contenu de la base de données

La base de données contient 155 327 mots organisés en 175 979 synsets pour un total de 207 016 paires mot-sens; sous forme compressée, sa taille est d'environ 12 mégaoctets.

WordNet inclut les catégories lexicales, noms, verbes, adjectifs et adverbes, mais ignore les prépositions, les déterminants et les autres mots de fonction[18].

.

[18]https://stringfixer.com/fr/Princeton_WordNet

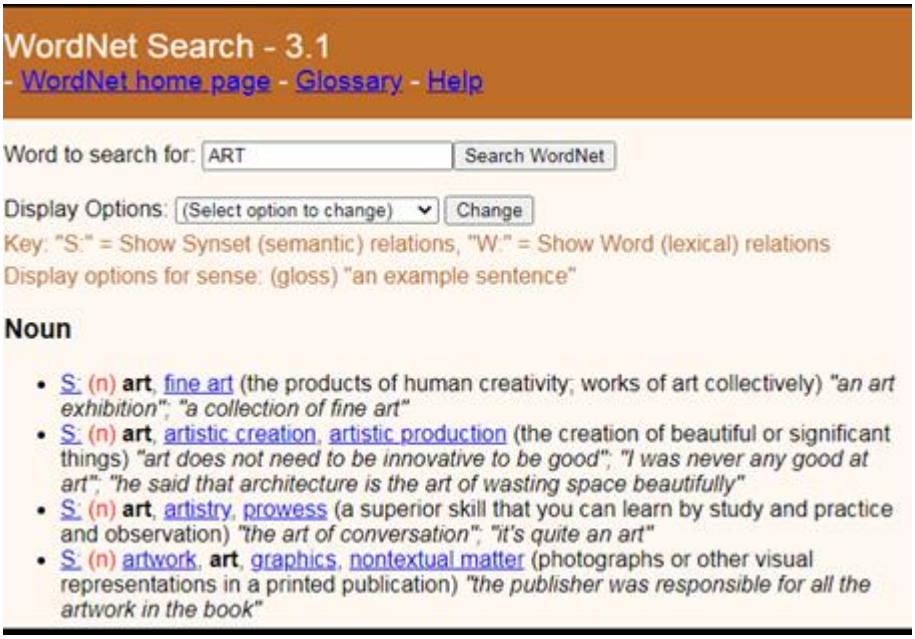


Figure 3.6 : Exemple manipulé avec le WordNet .

3- langages utilisés

3-1 XML ou (eXtensible Markup Language)

Est un langage informatique de balisage générique qui permet de décrire des données à l'aide de balises et de règles que l'on peut personnaliser.

L'objectif du XML est de faciliter les échanges de données entre les machines. A cela s'ajoute un autre objectif important : décrire les données de manière aussi bien compréhensible par les hommes qui écrivent les documents XML que par les machines qui les exploitent.

Le XML a été créé pour faciliter les échanges de données entre les machines et les logiciels.

4- Algorithmes utilisés

4-1 HUIM est un problème difficile car la fonction d'utilité pour la sélection de motifs n'est ni anti monotone ni monotone. Pour faire face à ce défi, les algorithmes HUIM s'appuient sur des bornes supérieures anti-monotones sur l'utilité pour réduire l'espace de recherche, telles que l'utilité pondérée par les transactions (TWU) et la borne supérieure de l'utilité restante. Les algorithmes HUIM peuvent être classés en deux catégories : les algorithmes basés sur deux phases et les algorithmes basés sur une phase.

Parmis les algorithmes existants pour la fouille des itemsets à haute utilité nous avons choisi TwoPhases.

4-2 TwoPhases :

Phase I:

Definition 1 :

(Transaction Utility) :

L'utilité transactionnelle de la transaction Tq, notée

Comme tu(Tq), est la somme des utilités de tous les items de Tq :

Tu(Tq)=$\sum_{ip \in Tq} \cup (ip, Tq)$**.**

Définition 2 :

(Utilisation pondérée par les transactions) :

L'utilisation pondérée par les transactions d'un itemset X, noté twu(X), est la somme des utilités de transaction de tous les transactions contenant X :

TWU(X)=$\sum_{x \subseteq Tq \in D} TU(Tq)$

Définition 3 : (High Transaction-weighted Utilization Itemset) Pour un itemset donné

X, X est un ensemble d'items à forte utilisation pondérée par les transactions

si twu(X) ≥ ε', où ε' est le seuil spécifié par l'utilisateur.

Ainsi, X est un ensemble d'éléments à utilisation pondérée par les transactions et

X ∈HTWU.

Phase II :

Dans la phase II, une analyse de la base de données est effectuée pour filtrer les ensembles d'éléments à haute utilité des ensembles d'items d'utilisation pondérés par transaction identifiés dans la Phase I. Le nombre les ensembles d'éléments d'utilisation pondérés par les transactions sont petits lorsque ε' est élevé. Dès lors, le temps Économisé dans la phase I peut compenser le coût encouru par l'analyse supplémentaire dans la phase II.

5- Etapes principales l'approche proposée

5-1 Entrées

l'entrée de notre système sont des fichiers sous format XML. Un exemple d'une partie d'un document XML du corpus SemEval est donnée par la figure …où chaque document est constitué de plusieurs textes. Chaque texte contient est constitué de plusieurs phrases contenant plusieurs mots. Chaque mot peut être un mot simple étiqueté par son lemma et sa catégorie syntaxique (nom, verbe, adjectif, adverbe,… etc), ou peut être un mot cible à désambiguïser (instance) étiqueté en plus d'un identifiant.

5-2 Prétraitement du contexte

Afin d'associer à chaque mot son utilité dans son contexte, nous avons utilisé KeyBERT pour extraire **les mots-clés** de chaque mot dans le fichier XML d'entrée XML. Si ce mot est un mot clé, un attribut **utility** est ajouté à sa description dans le fichier XML résulat avec comme valeur son poids*100.

Un extrait du document obtenu est donné par la figure (3.7) :

Figure 3.7 : Le Résultat Obtenu.

5-3 Wordnet est relations sémantiques

- Dans cette étape nous avons utilisé le WordNet pour extraire toutes les définitions possible pour chaque sens d'un mot cible donné. En plus, nous avons exploité quelques relations sémantiques selon le cas pour la construction des transactions associées à chaque sens. Parmi les relations sémantiques utilisés, nous citons :

Noms: Hypernymes, Hyponymes , Holonyme.

Verbes: Hypernyme ,Troponyme.

Les figures présentent les transactions obtenus pour les sens associés au mot cible « **art** ».

Figure 3.8 : Transactions Obtenue Pour Les Sens Associés Au Mot Cible « **Art**».

5-4 Extraction d'éléments à haute utilité HUIM

A partir de chaque liste de transactions obtenu pendant l'étape précédente, l'algorithme **TwoPhases** est appliqué pour extraire une liste des itemsets fréquents à haute utilité qui dépassent un certain seuil fixé par l'utilisateur (ex : min_utility=30) et dont la taille est fixée au paravent (Ex taille=2).

Ensuite, la similarité entre le mot cible et un sens sélectionné est calculé en utilisant la similarité de Wu & Palmer (Réf) en suivant la formule suivante :

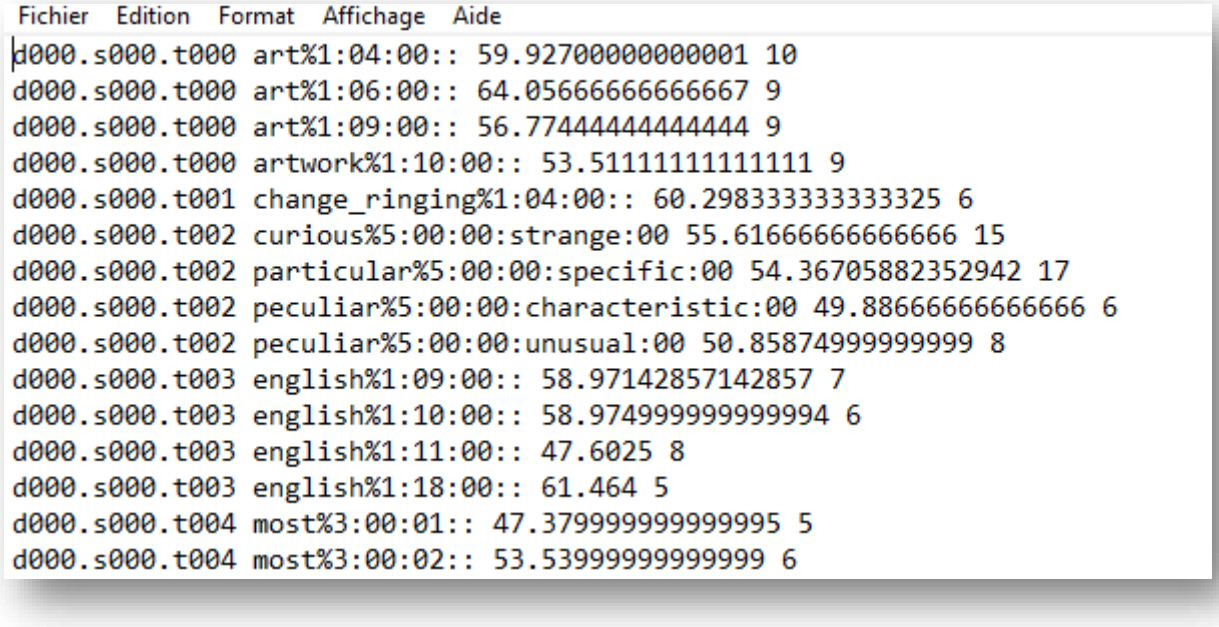
$$Sim(w, S_i) = \frac{1}{N}\sum_{k=1}^{N} \max\left(WuP\left(w, S_{i,k}^{1}\right), WuP(w, S_{i,k}^{2})\right)$$

Où :

- S_i : le sens sélectionné pour le mot cible w.
- $S_{i,k}$: le k-itemset de l'ensemble des HUIs obtenus.
- $S_{i,k}^{1}$ et $S_{i,k}^{2}$: le premier et le deuxième mot du $S_{i,k}$ respectivement.
- $WuP(S_1,S_2)$: la similarité de Wu & Palmer qui calcule la parenté en considérant les profondeurs des deux synsets dans les taxonomies WordNet, ainsi que la profondeur du LCS (Least Common Subsumer) selon la formule suivante :

$$WuP(S_1, S_2) = 2 * \frac{\text{depth}(LCS(S_1, S_2))}{\text{depth}(S_1) + \text{depth}(S_2)}$$

La figure (3.9) suivante présente un extrait des mesures de similarité obtenu pour chaque sens pour un mot cible.



```
Fichier  Edition  Format  Affichage  Aide
d000.s000.t000 art%1:04:00:: 59.92700000000001 10
d000.s000.t000 art%1:06:00:: 64.05666666666667 9
d000.s000.t000 art%1:09:00:: 56.77444444444444 9
d000.s000.t000 artwork%1:10:00:: 53.51111111111111 9
d000.s000.t001 change_ringing%1:04:00:: 60.298333333333325 6
d000.s000.t002 curious%5:00:00:strange:00 55.61666666666666 15
d000.s000.t002 particular%5:00:00:specific:00 54.36705882352942 17
d000.s000.t002 peculiar%5:00:00:characteristic:00 49.88666666666666 6
d000.s000.t002 peculiar%5:00:00:unusual:00 50.85874999999999 8
d000.s000.t003 english%1:09:00:: 58.97142857142857 7
d000.s000.t003 english%1:10:00:: 58.974999999999994 6
d000.s000.t003 english%1:11:00:: 47.6025 8
d000.s000.t003 english%1:18:00:: 61.464 5
d000.s000.t004 most%3:00:01:: 47.379999999999995 5
d000.s000.t004 most%3:00:02:: 53.53999999999999 6
```

Figure 3. 9 **:** Extrait des mesures de similarité obtenu pour chaque sens.

Enfin, le sens qui maximise cette similarité est considéré comme le sens approprié du mot cible.

$$WSD(w) = \max_i Sim(w, S_i)$$

5-5 Résultat final

Le résultat final de notre système est un fichier contenant la liste des mots cibles suivi d'un identifiant du synset correspondant au sens approprié. La figure (3.10) représente un extrait du fichier résultat.

```
d000.s000.t000 art%1:09:00::
d000.s000.t001 change_ringing%1:04:00::
d000.s000.t002 peculiar%5:00:00:characteristic:00
d000.s000.t003 english%1:10:00::
d000.s000.t004 most%3:00:02::
d000.s000.t005 english%3:01:01::
d000.s000.t006 peculiarity%1:09:00::
d000.s000.t007 unintelligible%3:00:00::
d000.s000.t008 respite%1:28:00::
d000.s000.t009 populace%1:14:00::
d000.s001.t000 tailor%1:18:00::
d000.s002.t000 england%1:15:00::
d000.s003.t000 scene%1:06:01::
d000.s003.t001 educe%2:36:00::
d000.s003.t002 rural%3:00:00::
d000.s003.t003 england%1:15:00::
d000.s003.t004 lovely%5:00:00:beautiful:00
d000.s003.t005 ancient%5:00:00:past:00
d000.s003.t006 rock%1:27:00::
d000.s003.t007 church%1:06:00::
d000.s003.t008 stand%2:42:03::
d000.s003.t009 field%1:14:00::
d000.s003.t010 sound%1:19:00::
d000.s003.t011 bell%1:06:00::
d000.s003.t012 cascade%2:35:00::
d000.s003.t013 tugboat%1:06:00::
d000.s003.t014 call%2:32:14::
d000.s003.t015 faithful%1:14:01::
d000.s003.t016 evening_prayer%1:10:00::
d000.s004.t000 parishioner%1:18:00::
d000.s004.t001 break%2:30:02::
d000.s004.t002 chew_the_fat%2:32:00::
d000.s004.t003 church%1:06:00::
d000.s004.t004 door%1:06:03::
d000.s004.t005 member%1:18:00::
d000.s004.t006 here%4:02:00::
```

Figure 3.10 : Extrait du fichier résultat.

Pour Evaluer les performances de notre approche, nous avons utilisé les mesures d'évaluation à savoir : le rappel, la précision et le F-mesure.

Précision : la proportion des items pertinents parmi l'ensemble des items proposés.

Précision= $\frac{TP}{TP+FP}$

Rappel: est la proportion des items pertinents proposés parmi l'ensemble des items pertinents.

$$\textbf{Rappel} = \frac{TP}{TP+FN}$$

F-mesure : Une mesure qui combine la précision et le rappel est leur moyenne harmonique, nommée F-mesure ou F-score

F-mesure =2* $\frac{précision*rappel}{précision+rappel}$

Les tableaux suivants présentent les résultats obtenus pour les différentes mesures de similarité en fixant le min utilité à 30 et en variant la relation sémantique utilisée pour tout le corpus et pour chaque catégorie syntaxique (nom, verbe, adjectif et adverbe).

Tableaux 3.1 **:** Résultats obtenus pour Les Différentes Mesures De Similarité Avec None.

POS	Précision	Rappel	F1
All	42,81%	42,81%	42,81%
Noun	56,00%	56,00%	56,00%
Verb	16,05%	16,05%	16,05%
Adj	45,62%	45,62%	45,62%
Adv	37,01%	37,01%	37,01%

Tableaux 3.2 **:** Résultats Obtenus Pour Les Différentes Mesures De Similarité Avec Exam.

POS	Précision	Rappel	F1
All	40,45%	40,45%	40,45%
Noun	52,44%	52,44%	52,44%
Verb	15,47%	15,47%	15,47%
Adj	44,04%	44,04%	44,04%
Adv	34,65%	34,65%	34,65%

Tableaux 3.3 **:** Résultats Obtenus Pour Les Différentes Mesures De Similarité Avec Hypernyme.

POS	Precision	Rappel	F1
All	42,99%	42,99%	42,99%
Noun	56,29%	56,29%	56,29%
Verb	16,25%	16,25%	16,25%
Adj	45,62%	45,62%	45,62%
Adv	37,01%	37,01%	37,01%

Tableaux 3.4 : Résultats obtenus pour des différentes mesures de similarité avec Hyponyme.

POS	Précision	Rappel	F1
All	41,76%	41,76%	41,76%
Noun	52,53%	52,53%	52,53%
Verb	18,57%	18,57%	18,57%
Adj	45,62%	45,62%	45,62%
Adv	37,00%	37,01%	37,00%

Tableaux 3.5**:** Résultats obtenus pour les différentes mesures de similarité avec la relation sémantique Holonyme.

POS	Précision	Rappel	F1
All	0,42901	0,42901	0,42901
Noun	0,562	0,561914	0,561957
Verb	0,160542	0,160542	0,160542
Adj	0,45618	0,45618	0,45618
Adv	0,37	0,370079	0,370039

Tableaux 3.6**:** Résultats obtenus pour les différentes mesures de similarité avec la relation sémantique Meronyme.

POS	Précision	Rappel	F1
All	0,424628	0,424628	0,424628
Noun	0,552533	0,552533	0,552533
Verb	0,160542	0,160542	0,160542
Adj	0,45618	0,45618	0,45618
Adv	0,37	0,370079	0,370039

Tableaux 3.7**:** Résultats Obtenus Pour Les Différentes Mesures De Similarité Avec La Relation Sémantique Entail .

POS	Précision	Rappel	F1
All	0,428133	0,428133	0,428133
Noun	0,560038	0,560038	0,560038
Verb	0,160542	0,160542	0,160542
Adj	0,45618	0,45618	0,45618
Adv	0,37	0,370079	0,370039

6- Conclusion

Les modèles actuels de traitement des langages naturelles 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 problèmes similaires, comment notre problème la désambiguïsation de sens des mots

.dans Ce chapitre, nous avons expliqué comment nous traitons des données textuelles volumineuses et sélectionnons un modèle qui convient à nos données. Ce modèle montré de bons résultats pour la désambiguïsation du sens des mots.

Comme les résultats final ilya des tableau utiliser de calculer la meseur les relation semantique .

Suivants présentent les résultats obtenus pour les différentes mesures de similarité

en fixant le min utilité à 30 et en variant la relation sémantique utilisée pour tout

le corpus et pour chaque catégorie syntaxique (nom, verbe, adjectif et adverbe).

Conclusion générale

Conclusion générale

Dans cette étude, nous avons passé en revue la tâche de désambiguïsation du sens des mots, DSM est la plus difficile

problèmes ouverts de la DSM. Certains des défis du DSM ont été discutés, de nombreux problèmes surviennent dans DSM parce qu'il dépend de connaissances tirées de différentes ressources. DSM traite également avec la complexité des langues. Approche basée sur les connaissances, supervisée, non supervisée approche et les approches semi-supervisées sont utilisées dans DSM. L'approche supervisée fonctionne ainsi que par rapport à toutes les autres approches car les données d'entraînement dépendent totalement de domaine.

Bibliographie

AMIA, S. (2022, mars 17). Les domaines d'application du NLP. *LinkedIn ttps://www.journaldunet.fr/* .

Bathelot, B. (2017, 07 12). Texte mining ou text mining. *definitions-marketing* . Récupéré sur http://www.definitions-marketing.com/Definition-Texte-mining-ou-text-mining

Besbes, A. (2021). *How to Extract Relevant Keywords with KeyBERT.* Récupéré sur https://medium.com/

Cannobio, d. P. (2021). Génération de langage naturel : ce que c'est, à quoi ça sert et comment apprendre à l'utiliser. *Contents* https://magazine.contents.com/.

Curry. (2002, octobre 11). *Wikipédia.* Récupéré sur https://fr.wikipedia.org/wiki/Compr%C3%A9hension_du_langage_naturel

Diksha Khurana, A. K. (2017, 8 17). Diksha Khurana, Aditya Koli, Kiran Khatter, Sukhdev Singh https://www.researchgate.net/.

DJEFFAL, D. A. (2014). Cours Fouille de données avancée. 96. Biskra, Département d'Informatique https://scholar.google.fr/ .

H. Cherfi. (2014). «Etude et réalisation d'un système d'extraction de connaissances à partir detextes» https://tel.archives-ouvertes.fr/.

Houria, D. (2012). *Classification des documents médicaux basée sur le Text Mining.* Blida : Université Blida 1 https://di.univ-blida.dz/.

lina. (2020, juillet 22). *data scientest*. (JPO) Récupéré sur JPO: https://datascientest.com/pages-presentation-formations

Majumder, P. (2021, June 24). Récupéré sur https://www.analyticsvidhya.com/blog/2021/06/word-sense-disambiguation-importance-in-natural-language-processing/

Mansour, K. (2021, 08 janvier). *Early Metrics.* Récupéré sur https://earlymetrics.com/fr

Oskar Wysocki, M. F. (2020, May). *ResearchGate Logo.* Récupéré sur https://www.researchgate.net/

Philippe, F. (2015, 04 18). The Data Mining Blog https://data-mining.philippe-fournier-viger.com/page/3/ .