République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieure et de la Recherche Scientifique

Université Abbes Laghrour –Khenchela-

Faculté des Sciences et de la Technologie

Département des Mathématiques et Informatique

Mémoire de fin d'étude, en vue de l'obtention du diplôme de Master en Informatique

Spécialité : Sécurité et Technologie Web

Thème :

Un démarche basée sur l'application des mesures de similarité dans la taxonomie WordNet pour le processus WSD (Word sens désambiguïsation)

Préparés par :

- TOBBI Aicha
- BOUCHAREB Romaissa

Encadré par :

- Dr. NESSAH Djamel

Année universitaire : 2022/2023

Remerciement

Avant tout nous remercions « ALLAH » de nous avoir donné la force et la patience d'attendre ce jour spécial et de mener à bien ce travail, nous le remercions du fond du cœur pour sa grande bénédiction qui nous a accompagée tout au long d notre parcours universitaire.

Nous remercions très sincèrement notre encadreur « Dr. NESSAH Djamel », qui nous a permis de bénéficier de son encadrement, et son soutien pour nous à chaque étape de notre travail, nous vous disons tous nos remerciements et notre appréciation. Nous remercions également les membres du jury pour son intérêt et son dévouement dans la correction et le suivi de ce mémoire.

Nous remercions nos parents pour leur soutien constant et continu, sans eux nous n'aurions pas réalisé ce dont nous rêvions autrefois.

Nous remercions très sincèrement notre collègue « ZEGGADA Omar » pour ses efforts pour nous aider.

Je dédier ce travail à :

- *A ceux qui ont travaillé dur pour moi et qui ont sacrifié leur vie au cours de mes études, ils m'ont tout donné de la plus belle des choses pour poursuivre mes études et m'ont encouragé à continuer jusqu'à ce que j'en sois là. Mon père et ma mère, je vous remercie de tout mon cœur pour vos efforts et vos sacrifices pour la réussite de ma carrière universitaire et de mon diplôme, et je vous dédie cette chère occasion.*
- *A mes sœurs Razane , Rihab et mon frère Saad.*
- *A mon adorable petite sœur Rama ,qui sait toujours comment procurer la joie et le bonheur pour toute la famille.*
- *À l'âme pure de ma grand-mère « Zahwa » dieu repose son âme.*
- *A tout les amies que j'ai connu jusqu'à maintenant*
- *sans oublier mon binôme Aicha pour son soutien moral, sa patience et sa compréhension tout au long de ce projet.*
- *A vous cher lecteur.*

-Romaissa-

Je dédier ce travail à :

- *A ceux qui ont travaillé dur pour moi et qui ont sacrifié leur vie au cours de mes études, ils m'ont tout donné de la plus belle des choses pour poursuivre mes études et m'ont encouragé à continuer jusqu'à ce que j'en sois là. Mon père et ma mère, je vous remercie de tout mon cœur pour vos efforts et vos sacrifices pour la réussite de ma carrière universitaire et de mon diplôme, et je vous dédie cette chère occasion.*
- *A mes sœurs, chacune en son nom, je remercie Dieu de votre présence dans ma vie.*
- *A mes chère frères « Mohammed » et « Djalel ».*
- *A mes cousines « Imane », « Hamza » et « Samira »et « Yaakoub».*
- *A ma chère amie « Manel », elle a toujours été mon refuge dans les moments de tristesse et de joie. En effet il n'y a pas de mots qui remplissent votre droit merci.*
- *A mon binôme « Romaissa » pour son soutien moral, sa patience et sa compréhension tout au long de ce projet.*

-Aicha-

Résumé

Le processus de désambiguïsation de sens des mots (WSD) est une tâche importante dans le traitement automatique du langage naturel. Qui consiste à attribuer un sens spécifique et correct à un mot dans un contexte donné. L'application des mesures de similarité est une méthode couramment utilisée pour le processus WSD. Les mesures de similarité sont utilisées pour calculer la distance entre les différents sens d'un mot et le contexte dans lequel il est utilisé. Les mesures de similarité les plus courantes sont basées sur WordNet, le sens le plus similaire au contexte est alors choisi comme étant le sens approprié.

ملخص:

تعتبر عملية إزالة الغموض عن المعنى (WSD) عملية مهمة في معالجة اللغة الطبيعية. والتي تتمثل في إسناد معنى محدد وصحيح لكلمة في سياق معين. يعد تطبيق مقاييس التشابه طريقة شائعة الاستخدام لعملية WSD، تُستخدم مقاييس التشابه لحساب المسافة بين المعاني المختلفة للكلمة والسياق الذي تستخدم فيه. تستند مقاييس التشابه الأكثر شيوعًا إلى WordNet ، ثم يتم اختيار المعنى الأكثر تشابهًا مع السياق باعتباره المعنى المناسب.

Abstract :

The word sens disambiguation (WSD) process is an important task in natural language processing. Which consists in attributing a specific and correct meaning to a word in a given context. The application of similarity measures is a commonly used method for the WSD process. Similarity measures are used to calculate the distance between different meanings of a word and the context in which it is used. The most common similarity measures are based on WordNet, the meaning most similar to the context is then chosen as the appropriate meaning.

Table des matières

Table des figures :

Liste des tableaux

Introduction générale

Grace aux nouvelles méthodes et techniques de l'informatique, les progrès enregistrés dans le développement d'outils de traitement de l'information, le monde connaît actuellement des avancées technologiques considérables et dans tous les secteurs.

Le traitement automatique de l'information en fait partie, c'est une approche permettant de comprendre la pensée humaine, de manipuler l'information, et l'exploiter pour les besoins de notre vie quotidienne.

Parmi les axes de traitement de l'information nous avons le traitement automatique du langage naturel, un domaine multidisciplinaire incluant la linguistique, l'informatique la sociologie, la psychologie cognitive et l'intelligence_artificielle. Le but est de développer des outils informatiques pour traiter automatiquement la langue naturelle, afin de répondre à des besoins comme : la traduction automatique, la comparaison des textes et détection de ressemblance (plagiat), la recherche d'information etc.

Cette discipline est née vers les années 1950, avec des tentatives de traduction automatique, ensuite vers 1954 une traduction complète de phrases en russe vers l'anglais a été réalisée avec succès.

Les années 70 – 80 ont marqué des tentatives de structuration de l'information pour la rendre compréhensible par la machine.

Les dernières avancées dans ce domaine ont trait avec l'intégration des technique de l'intelligence artificielle dans ces traitements, ainsi plusieurs concepts comme les ontologies, l'apprentissage automatique, les réseaux de neurones, les agents intelligents, la reconnaissance de formes ont propulsés considérablement les travaux liés à cette discipline.

Parmi les difficultés rencontrées lors des traitements des langages, nous avons le concept d'ambiguïté en général, c'est une caractéristique inhérente au langage naturel est un problème récurrent dans le domaine de traitement automatique de la langue. En effet, on peut rencontrer différents types d'ambiguïté selon le niveau d'analyse du langage dans lequel on se situe :

Au niveau lexical et syntaxique, au niveau sémantique, avec les différents sens des mots (homophones, polysémie, etc.). Les ambiguïtés constituent un majeur défi pour les systèmes automatisés de traitement du langage. Plusieurs recherches ont été menées pour résoudre ce problème, le processus est connu sous le nom de : désambiguïsation. [1]

C'est dans ce domaine que se situe le thème de ce mémoire, plus précisément, nous allons proposer une approche pour lever l'ambiguïté des termes dans un texte, en utilisant l'hiérarchie universelle WordNet.

Notre démarche vise a déterminer la proximité entre les sens possibles des différents mots dans un contexte donné, pour choisir le sens du mot ambigu le plus proche du voisinage, en se basant sur des

mesures de similarité entre des concepts hiérarchisés dans des ressources lexicales comme le dictionnaire WordNet.

Notre mémoire est divisée en quatre chapitres, le premier chapitre présenté le domaine du TALN, ses objectifs, son niveau de traitement, ses domaines d'application et ses difficultés. Le deuxième chapitre présente quelque définitions et les types d'ambiguïté, les sources linguistiques de l'ambiguïté, puis Les approches de désambiguïsation et la désambiguïsation dans les applications de TALN.

Ensuite le troisième chapitre présenté les mesures de similarité sa définition et ses utilisation, les mesures de similarité statiques et différentes méthodes du mesures de similarités sémantiques.

Enfin le quatrième chapitres c'est la Proposition de notre travaille nous parlons sur la taxonomie WordNet, choix de la mesure de similarité utilisée et l'approche de désambiguïsation proposée description de l'algorithme que nous avons utilisé et les outils d'implémentation.

Chapitre 01 : Le traitement automatique du langage naturel (TALN)

1. Introduction

Le but du traitement automatique du langage naturel est de créer des programmes informatiques capables de traiter automatiquement le langage naturel.

Les langues naturelles sont des langues parlées ou écrites par des humains, par opposition aux langues artificielles, informatiques, mathématiques ou logiques. En effet, le traitement ne concerne pas directement la langue, mais plutôt des données linguistiques, des textes, encodés dans une langue spécifique.

Les techniques de traitement automatique du langage naturel permettent d'extraire du texte des informations plus riches que de simples unités de vocabulaire. Ces informations morphologiques, syntaxiques et sémantiques ont été utilisées en partie dans la recherche d'informations (IR) pour améliorer les méthodes d'appariement, la représentation des documents et du contenu des requêtes, ainsi que le processus de recherche. [2] [3]

Ce chapitre présente l'objectif de TALN, les niveaux de traitement et les applications du TALN, ses différents formalismes de représentation de connaissances et finalement ses difficultés et les défis confrontés.

2. Bref historique

Le traitement automatique des langues est né à la fin des années quarante du siècle dernier dans un contexte scientifique et politique très précis. [4] [2]

✓ Zelling Harris publie ses travaux les plus importants de la linguistique (linguistique distributionnaliste) entre 1951 et 1954.

✓ En 1954 : la mise au point du premier traducteur automatique (très élémentaire). Quelques phrases russes, sélectionnées à l'avance, ont été traduites automatiquement en anglais. Bien que le vocabulaire ne comptait que 250 mots et la grammaire 6 règles, cette expérience a initié de nombreux travaux dans ce domaine.

✓ En 1956 : la naissance de l'intelligence artificielle à Dartmouth.

✓ N. Chomsky, a publié en 1957 ses premiers travaux importants sur la syntaxe des langues naturelles, et les relations entre grammaires formelles et grammaires naturelles.

✓ En 1962 : la première conférence sur la traduction automatique est organisée au MIT par Y.Bar-Hillel (Automatic Language Processing Advisory Council).

✓ Entre 1964 et 1966 : Joseph Weizenbaum écrit le programme ELIZA (est un programme qui simule des entretiens avec des psychiatres).

✓ En 1968 : le premier (vrai) système de traduction (Systran, russe- anglais).

✓ En 1972 : Terry Winograd, a créé le premier logiciel capable de dialoguer en anglais avec un robot.

✓ En 1976 : le système de traduction METEO (un système de traduction automatique conçu spécifiquement pour la traduction des bulletins météorologiques émis quotidiennement par Environnement Canada, ce système a été développé par John Chandioux).

✓ Dans les années 80 : système de reconnaissance statistique multi locuteur.

✓ Dans les années 90 : Premiers corpus, approches statistiques apprentissage automatique. Applications utilisent corpus de grande taille et méthodes statistiques.

✓ 2000s : Utilisation du World Wide Web comme corpus.

3. Définitions de « Traitement automatique du langage naturel »

- **Définition 1 :** On regroupe sous le vocable de traitement automatique du langage naturel (TALN) l'ensemble des recherches et développements visant à modéliser et reproduire, à l'aide de machines, la capacité humaine à produire et à comprendre des énoncés linguistiques dans des buts de communication. [5]
- **Définition 2 :** Le traitement automatique du Langage Naturel est un des domaines de recherche les plus actifs en science des données actuellement. C'est un domaine à l'intersection du Machine Learning et de la linguistique. Il a pour but d'extraire des informations et une signification d'un contenu textuel. [6]
- **Définition 3 :** Le Traitement automatique du langage naturel ou de la langue naturelle TALN, TAL est une discipline à la frontière de la linguistique, de l'informatique et de l'intelligence artificielle, qui concerne l'application de programmes et techniques informatiques à tous les aspects du langage humain. Ainsi, le TAL ou TALN est parfois nommé ingénierie linguistique. [7]

4. L'objectif de TALN :

L'objectif du traitement automatique du langage naturel (TALN) est de concevoir un logiciel capable de traiter automatiquement des données linguistiques, c'est-à-dire des données exprimées dans une langue (dite « naturelle »).

Ces données linguistiques peuvent être des textes écrits, des conversations écrites ou parlées, ou même des unités linguistiques plus petites que le texte normalement parlé (par exemple : des phrases, des déclarations, des phrases ou simplement des mots isolés). [8]

5. Les niveaux de traitement en TALN :

Nous introduisons dans cette section les différents niveaux de traitement nécessaires pour parvenir à une Compréhension complète d'un énoncé en langage naturel. Du point de vue de l'ingénieur, ces niveaux Correspondent à des modules qu'il faudrait développer et faire coopérer dans le cadre d'une application complète de traitement de la langue naturelle. [2]

- ❖ Niveau morphologique.
- ❖ Niveau syntaxique.
- ❖ Niveau lexical.
- ❖ Niveau sémantique.

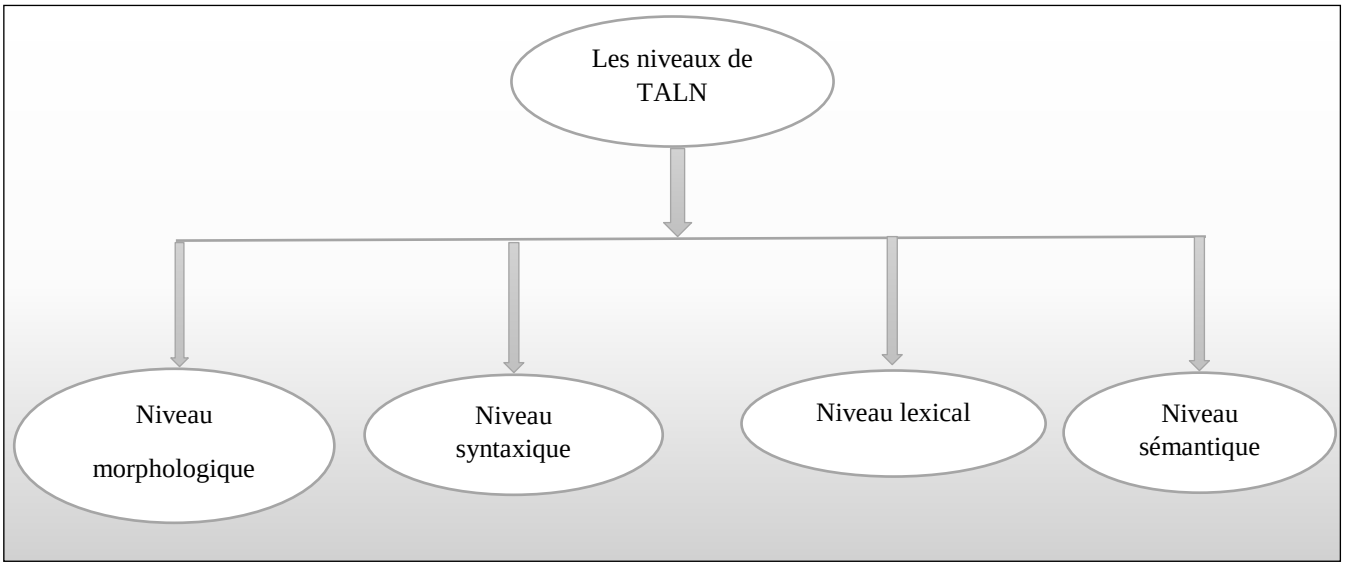

Figure 01: les niveaux de traitement du langage naturel. *[2]*

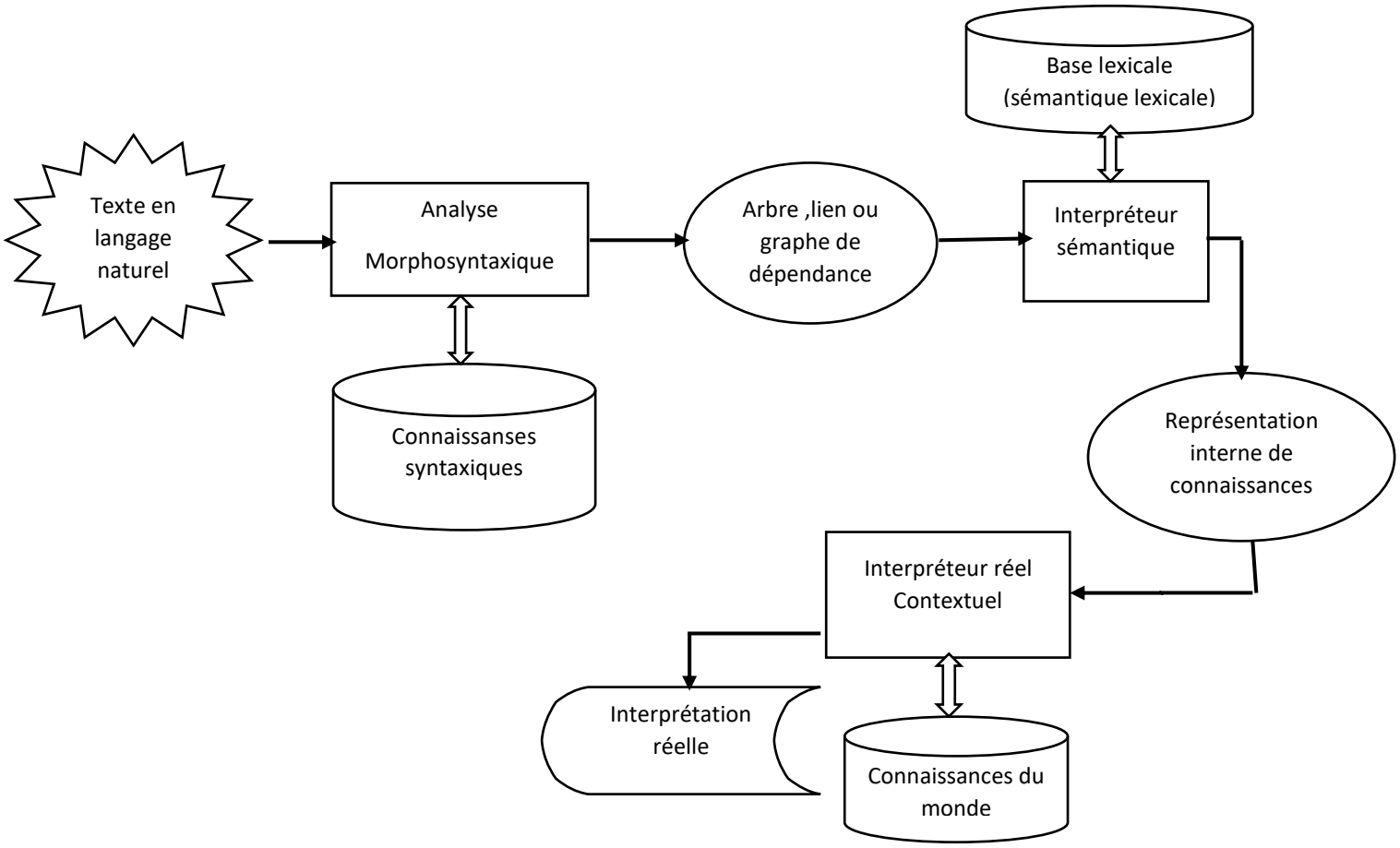

Figure 02: Architecture générale du TALN. *[9]*

5.1. Niveau morphologique :

La morphologie interprète comment les mots sont structurés et quels sont leurs rôles dans la phrase. Cette analyse consiste à une segmentation du texte en unités élémentaires auxquelles sont attachées des connaissances dans le système : une fois cette segmentation effectuée, ce n'est plus le texte qui est manipulé, mais une liste ordonnée d'unités. Pour le traitement d'un texte numérique : on part d'une chaîne de caractères typographiques, et on essaie de la segmenter de manière à ce que chaque partie corresponde à une unité classée dans le système. [2]

Exemple :

Soit la chaîne de caractères "عمر يكتب الدرس ":

La segmentation se fera de la manière suivante :

- **U1 = يكتب**
- **U2 = عمر**
- **U3 = الدرس**

Maintenant, on pourra associer toutes sortes d'informations aux **Ui** (i = 1, 2, 3, ...), comme :

➢ **U2 = عمر**

- Informations morpho-syntaxiques : nom propre, masculin, singulier.
- Informations sémantiques : animé, humain, prénom ...

➢ **U1 = يكتب**

- Forme lemmatisée : كتب
- Informations morpho-syntaxiques : verbe (فعل), passé (ماضي), indicatif, 3 ieme personne singulier .

5.2. Le niveau syntaxique :

C'est une partie de la grammaire qui traite la manière dont les mots peuvent se combiner pour former des propositions et de l'enchaînement des propositions entre elles. Cela consiste à associer, à la chaîne découpée en unités, une représentation des groupements structurels entre ces unités ainsi que des relations fonctionnelles qui unissent les groupes d'unités. [9]

5.2.1. Les constituants syntaxiques :

Les énoncés naturels ne sont pas simplement des séquences de mots, mais sont organisés en composants plus grands que le mot (les syntagmes) qui maintiennent des relations dominantes et de contrôle entre eux. Par conséquent, le deuxième objectif de l'analyse syntaxique est de relier chaque phrase à ses structures constitutives. L'organisation compositionnelle des énoncés est marquée de diverses manières par la prosodie dans la langue parlée (pauses, accent, montée ou descente mélodique, allongement de la syllabe finale, etc.). Elle est moins systématiquement transcrite au niveau de la figure par la ponctuation.

Un but important de l'analyse syntaxique est donc d'identifier les différents constituants et sous-constituants, ainsi que de repérer les relations que ces groupes entretiennent entre eux, et les fonctions syntaxiques qu'ils remplissent (sujet, objet direct, objet indirect, circonstant …). En d'autres termes, il s'agit d'associer à une séquence linéaire mono dimensionnelle d'unités lexicales une structure hiérarchique rendant compte des relations entre ces unités. [4]

5.2.2. L'arbre syntaxique :

Traditionnellement, le résultat de l'analyse syntaxique est représenté sous la forme d'un arbre, ce qui permet d'identifier simultanément les frontières de constituants, ainsi que les relations de dominance qu'ils entretiennent. [4]

Exemple : [2] Reprenons l'exemple précédant :"يكتب عمر الدرس" et sa représentation morphologique :

- **U1 = يكتب**

- **U2 = عمر**
- **U3 = الدرس**

U : Unité.

Le résultat de l'analyse syntaxique pourra être par exemple l'arbre suivant :

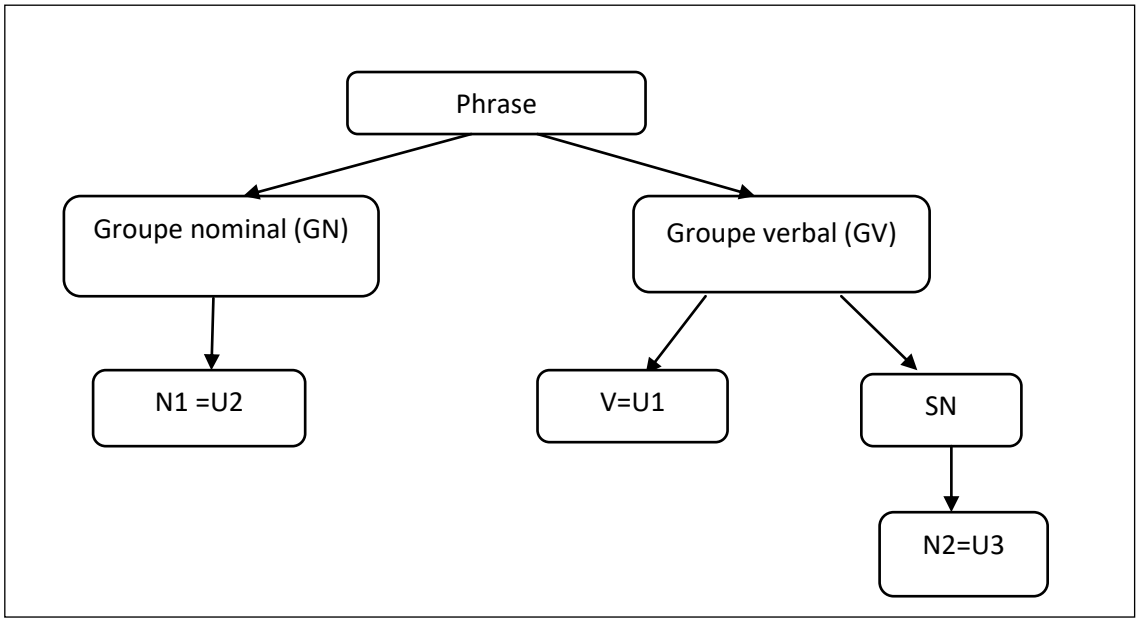

Figure 03: Arbre syntaxique de la phrase. *[2]*

Phrase = « يكتب عمر الدرس »

GN = عمر

GV = يكتب الدرس

SN (Sujet Nominal) = الدرس

N1 (Nom 1) = عمر

V (verbe) = يكتب

N2 (Nom 2) = الدرس

5.2.3. Difficultés de l'analyse syntaxique

La conception d'analyseurs syntaxiques fiables et rapides est un problème ardu. On est en effet confronté à une double contrainte : lutter contre la prolifération des ambiguïtés, tout en décrivant des phénomènes extrêmement complexes et subtils. Or, dans la pratique, ces deux contraintes sont largement contradictoires.

La réalité avec laquelle tout syntacticien doit composer d'emblée, c'est l'ambiguïté lexicale, qui fait que de très nombreuses formes graphiques correspondent à plusieurs entrées lexicales différentes, comme :

– souris : formes verbales de sourir, nom féminin singulier et pluriel.

– petit : adjectif ou nom masculin singulier.

– la : déterminant ou pronom personnel féminin singulier, nom masculin.

– mousse : formes verbales de mousser, nom masculin, nom féminin.

Si l'on se limite aux simples catégories syntaxiques de base, environ 50% des mots d'un texte sont ambigus, c'est-à-dire qu'ils correspondent possiblement à plusieurs catégories morpho-syntaxiques. Conséquence directe : une phrase de 20 mots a en moyenne 2^{10} interprétations différentes au niveau des étiquettes des feuilles l'arbre syntaxique. [4]

5.3. Le niveau lexical :

L'analyse lexicale fournit des informations sur le mot :

- informations grammaticales (nature du mot, ses flexions)

- informations sémantico-pragmatiques (traits, synonymes)

Cette analyse permet de lever l'ambiguïté lexicale : certains mots ont un sens différent selon le contexte ("suite","avocat"). La vocation des analyses morphologique et lexicale est de fournir les constituants de base des modules syntaxique et sémantique qui suivent.

Le but de cette étape de traitement est de passer des formes atomiques identifiées par le segmenteur aux mots, c'est-à-dire de reconnaître dans chaque chaîne de caractère une (ou plusieurs) unité(s) linguistique(s), dotée(s) de caractéristiques propres (son sens, sa prononciation, ses propriétés syntaxiques, etc.). [10] [3]

5.4. Le niveau sémantique :

La description et la formalisation au niveau sémantique est encore beaucoup plus complexe que les niveaux énoncés précédemment. De ce fait, très peu d'outils de traitement fonctionnent encore, ou du moins concernent des applications très limitées où l'analyse sémantique est restreinte à un domaine très restreint ; un analyseur sémantique généraliste complet, il reste encore beaucoup à apprendre.

Pour déterminer le sens d'une phrase, la première étape consiste à se concentrer sur le sens de chacun des mots qui composent la phrase. Ensuite, en utilisant les informations fournies par l'analyse syntaxique, le sens complet de la phrase peut être déduit à l'aide de la connaissance des relations existantes entre les mots.

Ces mots et ces structures constituent de nombreux indices pour le calcul du sens : On peut dire que le sens vient de la double dotation de sens et de sens relation entre les mots. [9] [3]

6. Les applications du TALN :

Les applications en TAL permettent de concevoir des interfaces de plus en plus adaptables à l'utilisateur d'une part, et d'autre part pouvant traiter (produire, lier rechercher, classer, analyser, traduire) de manière de plus en plus intelligente les informations disponibles sous forme textuelle. [11]

6.1. Le traitement documentaire :

Les applications du TAL visent à faciliter le traitement par l'humain des immenses ressources disponibles en langage naturel. Ces applications sont par exemple :

6.1.1. La traduction automatique :

Cette application, qui a historiquement suscités les premiers efforts de recherche en TALN, reste un enjeu économique et politique de première importance. Si de tels traducteurs existaient, il serait sans doute beaucoup moins crucial de recourir, pour assurer une large diffusion à des documents.

6.1.2. La recherche de documents :

Intéressants dans des bases documentaires. La prolifération des outils de recherche documentaire sur la toile, qui traitent quotidiennement des millions de requêtes, montrent bien l'importance de la demande en la matière. Les performances de ces moteurs témoignent du chemin qu'il reste à parcourir dans ce domaine. Si Google semble aujourd'hui sortir du lot, d'autres moteurs de recherche valent certainement la peine d'être connus.

6.1.3. L'analyse d'un corpus de documents relatifs à un thème donné (historique, économique, veille technologique) :

Une application typique de ce domaine consiste à fournir des outils de visualisation et d'exploration dynamique de champs disciplinaires (scientifiques).

6.1.4. Le mécanisme de reconnaissance des mots composés :

La mise en œuvre des grammaires régulières locales, qui vont détecter toutes les occurrences des patrons typiques de production des mots-composé. Ainsi en Français, ces grammaires détecteront les séquences N Adj (langage naturel), N PREP N (réseau de neurones, machine à écrire). [12]

6.2. Les interfaces naturelles :

Les interfaces naturelles constituent le dernier domaine d'application apparu en TAL elles concernent :

6.2.1. L'interrogation en langage naturel de base de données :

La mise en place de multiples applications de ce type commencent à se mettre en place sur la toi le. Comme traduction langage naturel vers SQL via ses interfaces ou de moteurs de recherche sur la toile.

6.2.2. Les interfaces vocales :

Qui mettent en œuvre de manière variable suivant les applications des modules de reconnaissance de parole, synthèse de parole, génération et gestion de dialogue, accès aux bases de connaissance,..., chacun de ces modules demandant des traitements spécifiques (désambiguïsation morphosyntaxique et identification de syntagmes pour la synthèse, grammaires stochastiques pour la reconnaissance de la parole...). [12]

7. Compréhension et formalisme de représentation :

La compréhension littérale d'un texte nécessite divers types de connaissances (modèle de la langue, modèle de la tâche, éventuellement état de la tâche, historique du dialogue et modèle utilisateur).

L'utilité de construire un module de compréhension nous donne l'avantage d'en extraire ce que nous appelons le « sens utile » d'un texte (informations nécessaires pour l'application). Si on situe la compréhension par rapport à un système de commandes, le sens utile permet de construire sa commande.

La représentation sémantique peut être vue comme la fonction de transformation d'une représentation primaire vers une autre représentation interprétable par le contrôleur de dialogue d'un système interactif.

Dans la littérature informatique, une multitude de formalismes de représentations sémantiques est proposée pour la représentation interne d'une phrase, afin d'en révéler le sens. Nous pouvons entre autre citer :

* Les logiques (la logique des propositions, la logique des prédicats ou la logique modale), par exemple, le démonstrateur AGS (Audiotel Guide des Services) du CNET utilise la logique du premier ordre pour représenter le sens d'un énoncé.

* Les graphes conceptuels (Sowa), appelés aussi réseaux sémantiques ou graphes de Sowa, ont été développés par Sowa. Les logiques et les graphes de Sowa sont surtout utilisés dans le domaine de l'ingénierie des connaissances linguistiques. [9]

7.1. Le sens et sa représentation :

Nous allons nous intéresser, dans cette section, aux principaux formalismes permettant la représentation interne d'une phrase, afin d'en dégager le sens. La représentation du sens d'un texte

ou d'un énoncé est donc la structure obtenue en sortie du module de compréhension. Une description de la logique, des graphes de SOWA, des structures de traits et des attributs seront explicités dans cette section. Les deux premiers sont surtout utilisés dans le domaine de l'ingénierie des connaissances linguistiques, les deux autres sont très courants da+ns les interfaces homme-machine.

Bon nombre de ces formalismes sont presque identique à la logique des prédicats du premier ordre. Le choix d'un formalisme est donc avant tout accrédité à l'expert pour exprimer ces connaissances et aux algorithmes d'interprétation utilisés. On peut aussi associer au sein d'un même système plusieurs formalismes.

7.2.Les logiques :

Plusieurs approches logiques ont vu le jour comme la logique des propositions, la logique des prédicats ou la logique modale. Notons qu'aucune logique n'a réussi à représenter une phrase de façon complète. Mais elles peuvent nous satisfaire comme dans le cas particulier des serveurs vocaux interactifs : comme le démonstrateur AGS (Audiotel Guide des Services) du CNET qui utilise d'ailleurs la logique du premier ordre pour représenter le sens d'un texte ou d'un énoncé.

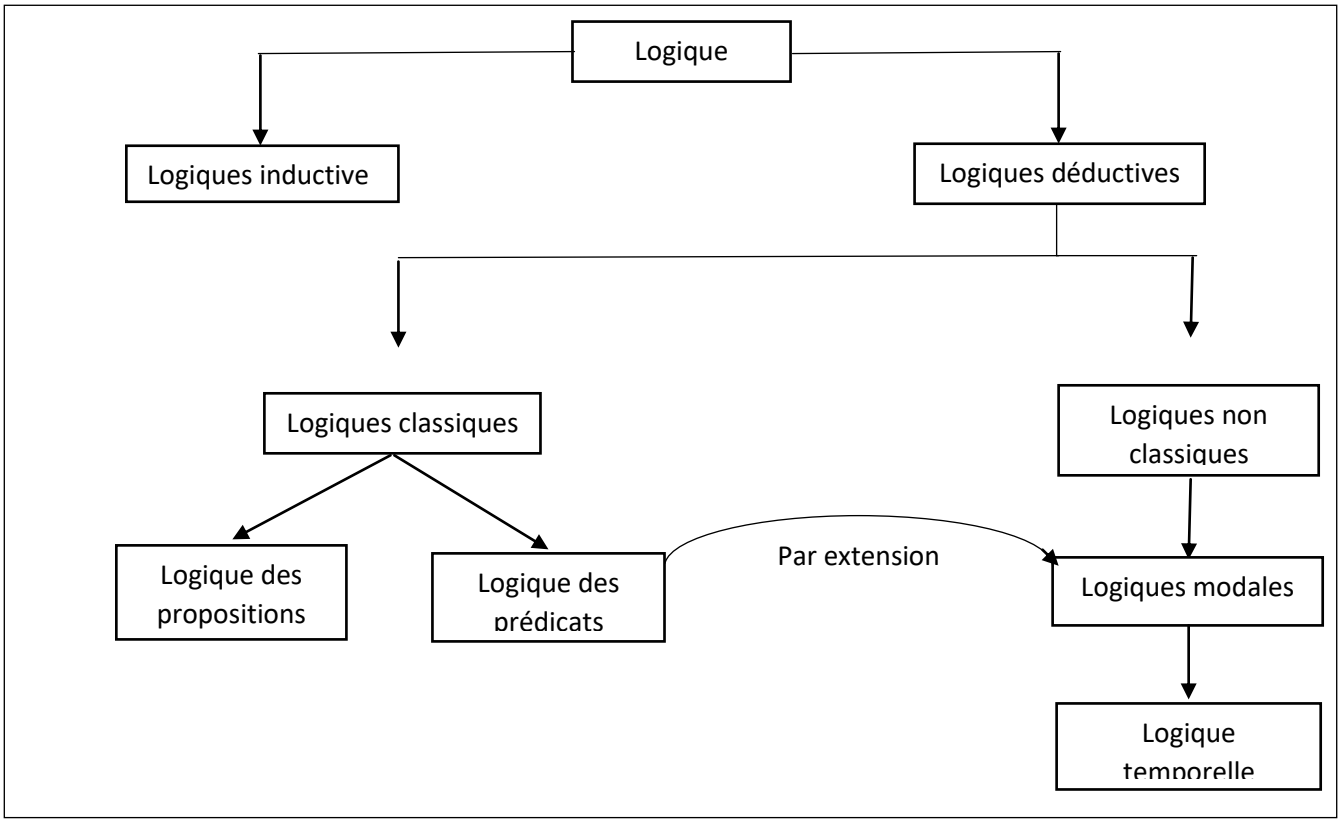

Figure 04: schéma des différents types de logiques. *[52]*

7.3. Les graphes conceptuels :

Ils on été proposées par Sowa (1984), afin de permettre de représenter la logique de manière graphique. Ces graphes sont plus formalisés que les réseaux sémantiques car leurs utilisations résident dans le traitement des langes naturels.

- **Définition :** Un graphe conceptuel simple est un graphe biparti, étiqueté, orienté et fini. Les deux composantes formant les nœuds du graphe sont : les concepts et les relations.
- **Un concept :** est composé d'un type et d'un référent :[<type> :<référent>], par exemple [Atelier : montage].
- **Le type de concept :**qui représente l'occurrence d'un événement classe d'objet Ils sont regroupé dans une structure hiérarchique nomé treillis de type concept.
- **Le référent :**préciser le sens du concept Occurrence du type de concept spécifié. ils peuvent sont de nature différente, notamment personnelles ou Génériques.

Les graphes conceptuels ou réseaux sémantiques, est une représentation graphique composé d'arcs orientés et de deux types de nœuds :

- Les nœuds représentant les entités (concepts) notés par des rectangles.
- Les nœuds représentant les relations notées par des ovales.

Les arcs relient deux nœuds de nature différente. Les entités sont définies par un type et un marqueur. Le marqueur peut désigner un objet en particulier (noté par le signe #, suivi d'un numéro référençant l'objet en question) ou au contraire un générique (noté par le signe *).

Les graphes conceptuels possèdent trois représentations possibles telles que, l'affichage graphique DF (Display Form), la lecture linéaire LF (Linear Form), et le CGIF (Conceptual Graph Interchange Format). [9] [13] [14]

Exemples : [13] phrase 1 : **«** Ali va a Constantine en bus **»**

Représentation de la phrase avec le Display Form (DF) :

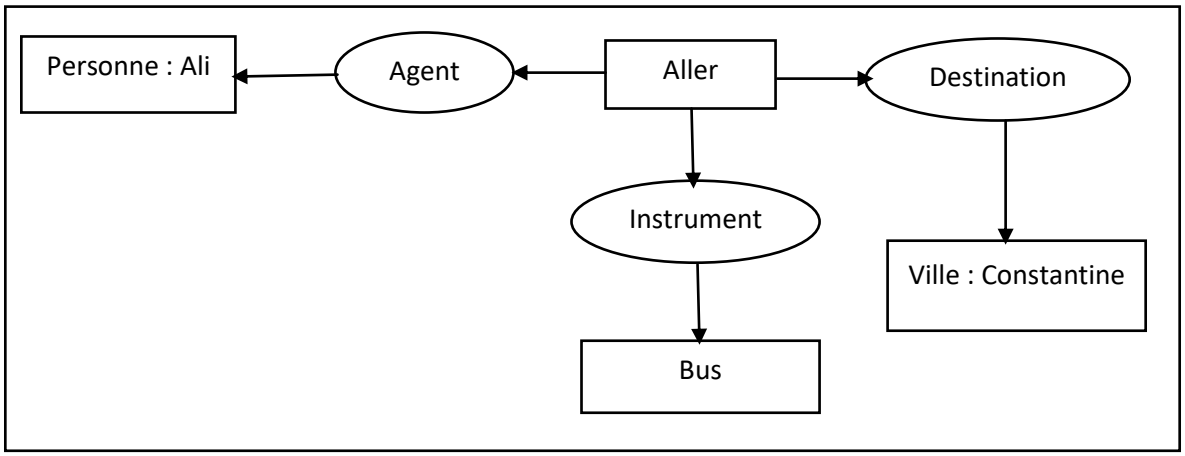

Figure 05: graphe conceptuel de la phrase 1 (DF). *[13]*

Dans cette exemple nous avons 4 concepts : [Aller], [Personne : Ali], [Ville : Constantine] et [Bus] et 3 relations conceptuelles : (Agent) relie [Aller] à l'agent Ali, (Destination) relie [Aller] à la destination Constantine, et (Instrument) relie [Aller] à l'instrument bus.

Avec la représentation LF (Linear Form) nous avons : [Aller]- (Agent)-> [Personne: Ali] (Destination)-> [Ville: Constantine] (Instrument)-> [Bus].

Avec la représentation CGIF form nous avons : [Aller : *x] [Personne: Ali *y] [Ville: Constantine*z] [Bus: *w] (Agent ?x ?y) (Destination ?x ?z) (Instrument ?x ?z).

- Un des intérêts de graphe conceptuel est que l'on peut très facilement ajouter des connaissances à un graphe : c'est le procédé de jointure de plusieurs graphes.

Exemples : [9] phrase 2 : « Le chaier1 rouge est sur la table ».

Phrase 3 : « Le chaier2 est sur la table ».

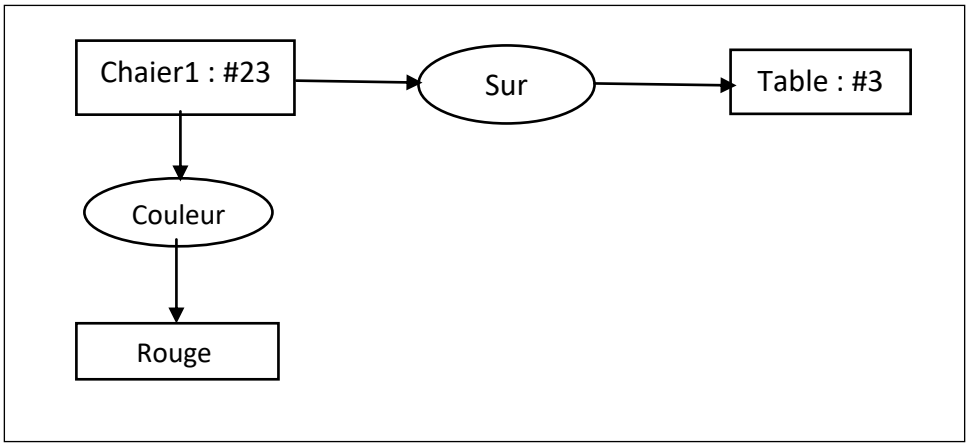

Figure 06: graphe conceptuel de phrase 1.

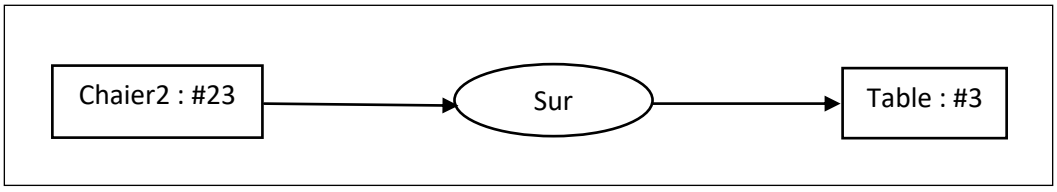

Figure 07: graphe conceptuel de phrase 2.

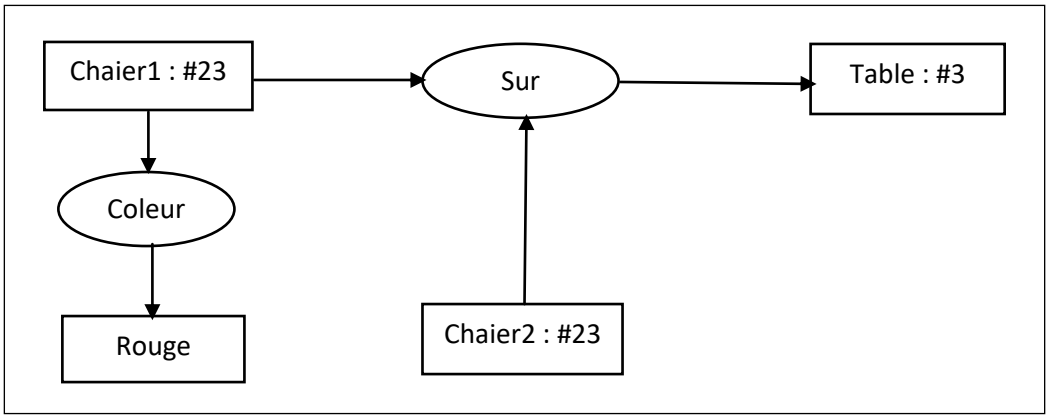

Figure 08: graphe conceptuel de la jointure de deux graphes.

7.4. Les frames :

- Le concept de « frame » était proposé par Minsky en 1975, l'dée est d'offrir un support permettant de regrouper l'ensemble d'information disponibles sur un objet (au sens large : concept, événement …).
- Le terme frame utilisé en synonyme avec (schéma).
- Le frame constitue d'un cadre dans lequel sont rassemblées les caractéristiques d'un objet.
- Un frame est un unité de connaissance (prototype) décrivant une situation ou un objet.
- Un frame posséde des attributs décrits par des facettes.
- Les facettes sont :

  -déclaratives (domaine, valeur, dfaut …).

  -procédurales (réflexes, démons …). [15]

Exemple1 :

Frame : Chaise.

Sort de : Meuble.

Nombre_de_pieds : doit étre entier par defult 4.

Style_du_dossier : doit étre Droit, Rembourré.

Nombre_de_bras : doit étre 0,1 ou 2.

Exemple 2 :

Frame : chaise_de_paul.

Sort de : chaise.

Nombre_de_pieds : 4.

Style_du_dossier : rembourré.

Nombre_de_bras : 0. [15]

8. Difficultés du TALN :

En traitement automatique des langages naturels (arab, français, anglais…), Les difficultés que l'on rencontre sont principalement de deux ordres, et ressortent soit de l'ambiguïté du langage, soit de la quantité d'implicite contenue dans les communications naturelles. [12]

8.1. L'ambiguïté :

L'ambiguïté est la qualité ou l'état de quelque chose qui laisse planer un doute sur l'interprétation, est quelque chose qui peut avoir plus d'une interprétation elle peut ressembler aussi à de l'indécision, de l'inexactitude, de l'incertitude.

L'ambiguïté peut se situer au niveau des mots, des expressions ou des phrases entières. Elle peut passer inaperçue dans les textes à contenu littéraire, poétique ou humoristique, mais elle doit être évitée dans les textes scientifiques ou journalistiques.

En grammaire, l'ambiguïté est une double interprétation due à une structure de phrase incorrecte. Le rôle de l'ambiguïté est d'impliquer des significations différentes pour le même message. Elle est également un mot qui désigne le manque de clarté d'une expression. [16]

Les ambiguïté peuvent toucher le mot comme la phrase, l'ambigüité portant sur le mot relève de l'homonymie, le récepteur devant identifier une forme, ou de la polysémie le récepteur devant choisir une signification parmi des plusieurs.

Dans les deux cas, l'ambiguïté ne vient pas du discours mais du langage, qui ne peut trancher entre des formes de signification qui s'excluent mutuellement. L'ambiguïté qui peut être définie comme la réduction de l'ambiguïté à la division de l'ambiguïté, n'est généralement que virtuelle et le contexte est généralement non ambigu. [17]

Il sera traité plus clairement dans le deuxième chapitre.

8.2. L'implicite :

L'activité linguistique se déroule toujours dans le contexte de deux interactions humaines et doit avoir une connaissance du monde et des fonctions, de sorte que la grande majorité des éléments contextuels nécessaires à la désambiguïsation et à la compréhension des énoncés naturels peuvent rester implicites. Une fois que les machines essaient de s'insérer dans le processus naturel de communication avec les humains, la situation change complètement : les machines n'ont pas cette connaissance de base, et si vous n'avez pas de base de connaissances, il est difficile, voire impossible, de comprendre pleinement la plupart des déclarations. [18]

9. Conclusion :

Dans ce chapitre, nous avons présenté le domaine du TAL, ses objectifs, son niveau de traitement, ses domaines d'application et ses difficultés. Les langues naturelles (arabe, français…) se caractérisent par la variabilité, l'intelligibilité et l'ambiguïté, posant de nombreux défis dans divers domaines tels que le traitement automatique du langage naturel ou la recherche d'informations.

Les entreprises peuvent tirer parti du traitement du langage naturel pour améliorer l'efficacité du traitement des documents, augmenter la précision des documents et identifier les informations les plus pertinentes à partir de grandes bases de données.

Chapitre 02 : la désambiguïsation dans les applications TALN

1. Introduction :

Dans le système de traitement automatique du langage naturel, l'ambiguïté est l'une des plus grandes difficultés rencontrées. L'ambiguïté fait partie intégrante du langage naturel, mais les humains en ont la capacité, et dans la plupart des cas avec l'aide du contexte, l'ambiguïté peut être résolue sans trop d'effort. Trouver des moyens de donner aux mots le bon sens est donc un contexte important. [19]

Afin de lever l'ambiguïté, de nombreuses méthodes efficaces ont été proposées. La désambiguïsation jouée un rôle très important dans les applications de le traitement automatique des langages naturel.

Dans ce chapitre, nous allons introduire une définition de l'ambiguïté et ses différents types, puis passer aux les approches de désambiguïsation et les travaux connexes outre l'application de la désambiguïsation dans les applications de traitement automatique des langues naturelles.

2. Définition de l'ambiguïté :

Les linguistes ont défini l'ambiguïté dans différentes définitions, notamment :

-Catherine FUCHS a défini l'ambiguïté comme : « (a) un cas de non bi-univocité entre formes et sens, (b) qui donne lieu à un choix nécessaire et impossible, et (c) qui constitue cas d'univocité dédoublée ». [20]

- D'autre part, Paul GRICE a défini l'ambiguïté avec la définition suivante : « elle représente une caractéristique des signes ou des structures syntaxiques de la langue (d'une langue particulière) qui menace potentiellement le succès des échanges communicatifs ».

-Une notre définition ; celle de JAKOBSON : « L'ambiguïté est une propriété intrinsèque, inaliénable, de tout message centré sur lui-même ».

-Le dictionnaire de l'linguistique LAROUSSE donne la définition de l'ambiguïté comme suit : « l'ambiguïté est la propriété de certaines phrases qui présentent plusieurs sens ».

Le dénominateur commun entre ces définitions mentionnées ci-dessus est que l'ambiguïté, et la définition de FUCHS est la plus détaillée, en ce qui concerne le terme « ambiguïté ».

Cette définition qui signifie que l'ambiguïté est un cas d'incompatibilité entre la forme et le sens c'est-à-dire lorsqu'une même forme peut avoir plusieurs sens, et ce cas nous guide à faire un choix nécessaire entre les différents sens et qui est impossible à le faire, et qui va finir par des solutions du même niveau. [21]

3. Les différents types d'ambiguïté :

3.1. Ambiguïté lexicale :

Ce type se produit lorsqu'un mot fait référence à plusieurs significations. Depuis les travaux de Saussure (1916), les linguistes s'accordent à dire qu'un mot (aussi appelé « signe ») est une association conventionnelle d'un signifiant et d'un signifié, le signifiant étant la forme physique (phonétique et orthographique) du mot, et le signifié, le contenu sémantique évoqué par ce signifiant.

De ce point de vue, un mot est considéré comme ambigu lorsqu'un signifiant correspond à plusieurs signifiés. C'est le cas des homonymes et des mots polysémiques, comme "souci" qui peut signifier à la fois une fleur et un souci, et l'exemple de "fer" qui peut signifier un métal ou un objet métallique. [21]

L'ambiguïté lexicale est la possibilité d'interpréter de plusieurs manières une phrase parlée ou écrite, ce qui en complique la compréhension, voire la rend impossible en l'absence d'informations complémentaires.

L'ambiguïté lexicale est souvent opposée à l'ambiguïté structurelle ou syntaxique, par laquelle c'est la construction de la phrase ou la place des mots qui rend difficile l'interprétation du discours écrit ou oral. L'ambiguïté lexicale, qui comprend ces deux catégories ainsi que d'autres, pose un problème notamment aux programmes de traitement automatique des langues (TALN). [22]

3.2. Ambiguïté grammaticale :

Ce type se produit lorsque la proposition ou la syntaxe d'un passage ne peut être déterminée sans un contexte très précis. Par exemple : dans la phrase « Jean envoie un vase de Chine » (Fuchs, 1996), on peut comprendre « Jean envoie un vase de Chine » ou « Jean envoie un vase fabriqué en Chine », ou « Sophie sent le rose" Dans cette phrase (Bodson, H. 2011), nous ne pouvons pas comprendre quel parfum floral respire Sophie, ou le parfum parfumé à la rose que Sophie porte. [21]

3.3. Ambiguïté syntaxique :

Une ambiguïté syntaxique apparait lorsque la structure de la phrase pourrait amener plusieurs sens. Les études sur les ambiguïtés syntaxiques tiennent rarement compte de l'importance du contexte.

L'ambiguïté syntaxique, qui apparait lorsqu'un syntagme a la possibilité d'avoir plusieurs points d'attache, ce qui peut conséquemment donner plusieurs sens à la phrase. [23]

L'ambiguïté syntaxique apparait lorsque le syntagme ou bien la structure de la phrase pourrait avoir plusieurs et différents signification. [21]

3.4. Ambiguïté pragmatique :

L'ambiguïté pragmatique, qui peut relever de deux niveaux différents, le calcul de valeurs de référence (par exemple la phrase qu'elle a écrite pourrait renvoyer à la situation actuelle - elle écrit - ou à la propriété actuelle - elle est l'écrivain : l'ancre référentielle de dont le processus est ambigu) ou le calcul de valeurs intermédiaires (ce qui pose la question de savoir si la source de l'énoncé est responsable de toutes les informations véhiculées par l'énoncé: il doit alors reconstituer la cible sous-jacente de l'énoncé). [17]

3.5. Ambigüité structurelle :

L'ambiguïté structurelle fait référence aux différentes interprétations possibles d'une déclaration écrite ou orales en raison de la manière dont les mots ou les phrases sont disposés. L'ambiguïté linguistique rend difficile pour un humain ou un système d'IA (comme un programme NLP) de comprendre le sens d'un énoncé jusqu'à ce qu'il dispose de plus d'informations pour clarifier le contexte.

L'ambiguïté structurelle des termes est souvent comparée à l'ambiguïté lexicale, généralement due au fait que les mots peuvent avoir plusieurs sens. Dans les deux cas, on parle d'ambiguïté linguistique, qui tient aussi à d'autres facteurs, dont le langage figuré et l'imprécision. [24]

L'ambiguïté structurelle résulte d'un mauvais placement des mots dans un énoncé, ce qui provoque une incompréhension de leur signification. [25]

4. Les sources linguistiques de l'ambiguïté :

4.1. L'homonymie :

Représentent le caractère des mots qui se prononcent de la même façon mais qu'ils renvoient à des sens différents, on distingue :

- les homophones non-homographes, qui ont la même prononciation, mais s'écrivent différemment. Par exemple le mot [t 3] peut s'écrire «teint », « tint », «thym », « tain ».
- les homophones homographes, qui partagent non seulement la même forme sonore, mais aussi la même forme graphique. Ces mots ont le même signifiant, alors qu'ils n'ont apparemment aucun sème commun. C'est le cas du mot « Palais », dont la forme actuelle est dérivée de deux mots distincts en latin« Palatium » pour l'acception « château » et « Palatum » pour la structure anatomique. [21]

4.2. La polysémie :

Le fait qu'un mot possède plusieurs significations s'appelle également polysémie. Le mot "grève", par exemple, peut désigner le sable de la terre ou bien, il peut s'agir de la cessation du travail pour un motif quelconque. Si le mot grève n'est pas utilisé dans son contexte, il peut

devenir source d'ambiguïté. En effet, la polysémie peut devenir responsable d'ambiguïté si le contexte d'énonciation n'est pas clair.

Dans la phrase "Ce vol était spectaculaire…", seul le contexte d'énonciation permet de la désambiguïser, c'est-à-dire d'apporter des informations qui permettront de savoir s'il s'agit du vol d'avion ou d'u cambriolage. [25]

5. Les approches de désambiguïsation :

5.1. Représentation vectorielle de sens pour la désambiguïsation lexicale à base de connaissances :

Loïc Vial, Benjamin Le couteux et Didier Schwab (2017) ont proposé une nouvelle méthode pour représenter les significations du dictionnaire sous forme vectorielle. Ils prennent les termes utilisés dans la définition, les projettent dans un espace vectoriel, puis additionnent les vecteurs résultants, en les pondérant en fonction de leur partie du discours et de leur fréquence. Les vecteurs de sens résultants sont ensuite utilisés pour trouver des significations pertinentes, permettant la création automatique de réseaux lexicaux. Le réseau résultant est ensuite évalué par rapport au réseau lexical WordNet construit manuellement. À cette fin, ils ont comparé l'impact de différents réseaux sur les systèmes de désambiguïsation basés sur la métrique « Lesk ». L'avantage de leur méthode est qu'elle fonctionne pour n'importe quelle langue sans réseau lexical, comme WordNet. Les résultats montrent que leur réseau généré automatiquement améliore le système de base, atteignant presque la qualité du réseau de WordNet.

Création de vecteurs de sens :

Leur modèle de représentation des significations du dictionnaire sous forme vectorielle est basé sur des définitions et des formes vectorielles de mots préexistantes. En pratique, la méthode est similaire à celle proposée par Ferrero et al, (2017) construire des vecteurs de phrases pour découvrir le plagiat inter linguistique. Leurs vecteurs de sens sont calculés comme la somme naturelle de tous les vecteurs de termes dans la définition de sens spécifiée. Ces vecteurs sont pondérés selon leur partie du discours (Part Of Speech, ou POS) : nom, verbe, adjectif ou adverbe, et aussi selon leur (Inverse Document Frequency ou IDF) : l'inverse du nombre de leurs occurrences dans l'ensemble dictionnaire. Plus formellement, nous avons :

- D(S) = {w0, w1, w2, . . ., wn} la définition du sens S dans le dictionnaire.
- Pos (wn) = {n, v, a, r} la partie du discours du terme wn (nom, verbe, adjectif, ou adverbe).
- Weight (pos) le poids associé à une partie du discours.
- Idf (wn) la valeur IDF de wn.

La définition du vecteur du sens S, notée $\emptyset(S)$ correspond à :

$$\emptyset(S) = \sum_{i=0}^{n} \left(\emptyset(w_n) \times weight\left(pos(w_n)\right) \times idf(w_n)\right)$$

Ils ont créé cinq modèles vectoriels de sens, tous basés sur le vocabulaire et les définitions de WordNet 3.0, mais avec différents modèles de vecteurs de mots. Les cinq modèles utilisées sont :

1. Un modèle pré-entraîné et proposés par Mikolov et al.(2013). Ce modèle a été entraîné sur environ 100 milliards de mots issus du corpus de nouvelles de Google. La taille du vocabulaire est d'environ de 3 millions de mots et les vecteurs ont une dimension de 300.
2. Un modèle de Pennington et al. (2014) GloVe, entraîné sur 42 milliards de mots issus de Common Crawl. Le vocabulaire est de 2 millions de mots et les vecteurs ont une taille de 300.
3. Un modèle de Levy & Goldberg(2014). L'apprentissage a été effectué sur Wikipedia, le vocabulaire est de 175 000 mots et la taille des vecteurs 300.
4. Le meilleur des modèles de prédiction de Baroni et al. (2014). La taille du vocabulaire est de 300 000 et la taille des vecteurs est de 400.
5. Et finalement, le meilleur modèle à base de comptage également créé par Baroni et al. (2014) de dimension 500 et une taille de vocabulaire identique au précédent modèle.

Ces modèles sont ensuite utilisés comme un réseau lexical pour améliorer le système basé sur DL (Désambiguïsation Lexicale)sur l'algorithme de Lesk. Ils ont utilisé une méthode similaire à celle de Banerjee & Pedersen (2002) aveccomme une grande différence dans l'utilisation d'une grille lexicale entièrement construite automatiquement.

Algorithme de désambiguisation lexicale :

Le système de DL que nous proposons est composé de deux éléments : un algorithme local qui calcule un score de similarité pour une paire de sens, et un algorithme global qui va chercher la meilleure combinaison de sens à l'échelle du document, en utilisant l'algorithme local.

L'algorithme local est l'élément central de leur système et c'est pour l'amélioration de ce dernier qu'est utilisé le réseau lexical. Comme système étalon ils ont utilisé une mesure de Lesk standard. Cet algorithme retourne, comme mesure de similarité, le nombre de mots communs à deux définitions de sens. Formellement, alors la mesure de Lesk entre deux sens S1 et S2 notée Lesk (S1, S2) est la suivante :

$$lesk(S1, S2) = |D(S1) \cap D(S2)|$$

Tel que : $D(S) = \{w_1, w_2 \ldots\ldots w_n\}$ la définition de S.

La mesure de Lesk étendue entre les sens S1 et S2 notée ExtLesk (S1, S2) est la suivante :

$$ExtLesk(S1, S2) = \left|\left(D(S1) \cup_{r \in rel(S1)} D(r)\right) \cap \left(D(S2) \cup_{r \in rel(S2)} D(r)\right)\right|$$

Tel que : rel(S) l'ensemble des sens reliés à S à travers un lien explicite dans WordNet.

Les résultats obtenus dans les deux tâches DL montrent une amélioration systématique de Score par rapport à la mesure Lesk classique (d'environ +3% à +9% selon les modèles).

Ils sont appliqué leur méthode uniquement sur la base lexicale WordNet pour évaluer nos modèles sur des tâches très connues en DL sur la langue anglaise. Cependant, leur méthode peut être appliquée très facilement à d'autres langues où les ressources sont moindres : un dictionnaire classique ainsi que des corpus non annotés sont les ressources suffisantes pour ainsi créer un système de désambiguïsation lexicale performant. [26]

5.2. Approches d'analyse distributionnelle pour améliorer la désambiguïsation sémantique :

Mokhtar Boumedyen Billami et Núria Gala(2016)ont proposé deux approches basées sur l'analyse de distribution pour réduire la complexité exponentielle en sélectionnant les voisins de distribution les plus proches sans perdre de cohérence au niveau de la désambiguïsation. Ils ont comparé les options du voisin distribué et du voisin le plus proche linéaire.

La clé de leur méthode de désambiguïsation est de sélectionner la distribution des plus proches voisins pour chaque polysémie dans le texte.

Leur méthode de désambiguïsation sémantique prend en compte des critères distributionnels. Les expériences qu'ils ont menées ont été menées sur des corpus anglais. Les expérimentations qu'ils proposent ne reposent pas seulement sur une seule méthode distribuée, mais également sur un large corpus d'évaluation, leur permettant ainsi de valider leur méthode.

Ils ont utilisé le corpus Europarl, le corpus parallèle des travaux du Parlement européen (Koehn, 2005) comme un corpus de travail. Pour valider leurs résultats sur le français, ils ont choisi ce corpus car il s'agit d'un corpus parallèle. Ils ont utilisé MateTools2 (Bohnet et Nivre, 2012) pour effectuer la lemmatisation, l'annotation des parties du discours et extraire les dépendances syntaxiques sur le corpus. Le corpus SemCor (Miller et al, 1993) est le corpus utilisé pour tester e tévaluer leurs méthodes.

BabelNet (Navigli et Ponzetto, 2012) est un réseau sémantique multilingue permettant de fournir des sens de mots, ils ont choisi d'utiliser BabelNet comme base de connaissances au lieu de WordNet parce qu'il propose plus d'informations sur les sens provenant de différentes ressources (Wikipédia, Wiktionnaire, Wikidata, Omega wiki, Open Multilingual WordNet) y compris WordNet. Dans ce travail, ils ont considé les sens comme étant des concepts et ils ne tenont pas

compte de la présence des entités nommées. BabelNet propose un mapping avec les sens de la version 3.0 de WordNet, ils ont donc utilisé la version 3.0 de SemCor.

Ils ont utilisé la méthode proposée par Lin (1998) à partir des dépendances syntaxiques extraites automatiquement depuis leur corpus de travail. Ces dépendances sont stockées et indexées. Ils ont à disposition un ensemble de relations grammaticales de dépendances syntaxiques. Cet ensemble ils permettent de mesurer le degré de cooccurrence entre deux mots. La similarité distributionnelle entre deux mots est définie par la fonction suivante :

$$Sim(w_1, w_2) = \frac{2x\ I\big(F(w_1) \cap F(w_2)\big)}{I\big(F(w_1)\big) + I\big(F(w_2)\big)}$$

$F(w_1)$ et $F(w_2)$ représentent l'ensemble des traits syntaxiques possédés respectivement parw_1 et w_2. $F(w_1) \cap F(w_2)$représente l'ensemble des traits syntaxiques communs entre w1 et w2.

Ils ont utilisé l'algorithme de Leak (1986) et ses variantes pour mesurer la similarité sémantique. Ces algorithmes nécessitent un dictionnaire (BabelNet dans leur cas) et aucun apprentissage. La fonction utilisée pour mesurer la similarité sémantique se présente par :

$$lesk(S1, S2) = |D(S1) \cap D(S2)|$$

Si aucune définition n'est proposée pour le sens, ils considèrent des synonymes. D'autre part, ils utilisent une variante de l'algorithme de Lesk (Navigli, 2009) qui consiste à comparer chaque sens candidat avec le contexte du mot w à désambiguïser. Comme contexte, ils ont considéré des phrases dans lesquelles la polysémie se produit. La fonctionnalité utilisée est fournie par :

$$lesk_{variante} = \big|context(w) \cap D\big(S_i(w)\big)\big|$$

Ils ont pu voir leur méthode comme un processus en deux étapes : la première utilise la similarité distributionnelle pour sélectionner les voisins les plus proches, et la seconde utilise la similarité sémantique pour lever l'ambiguïté. La similarité distributionnelle entre le mot à désambiguïser et chaque voisin sélectionné est plus forte que la similarité distributionnelle entre le mot à désambiguïser et tous les autres mots du contexte.

Ils ont adopté une approche structurelle basée sur la distance sémantique entre les significations, suivant la formule proposée par (Navigli, 2009) :

$$S^* = argmax_{S \in sens(w)} \sum_{N_i \in N_w :\, N_i \neq w} max\ Score(S, S')$$

Tel que : $S' \in Sens(N_i)$ $avec\ i = 1 \dots k$ $et\ N_w = \{N_1, N_2 \dots N_k\}$est l'ensemble ordonné des k voisins les plus proches du mot cible $w.Sens(N_i)$est l'ensemble des sens du voisin N_i et $sens(w)$est l'ensemble des sens du mot cible w. $Score(S, S')$est la fonction utilisée pour mesurer la similarité entre deux sens S et S' .

Ils utilisent comme contexte la phrase dans laquelle apparaît le mot à désambiguïser. En cas d'égalité des scores entre plusieurs sémantiques candidates, ils utilisent un algorithme heuristique qui considère la sémantique avec le plus de connexions sémantiques dans le réseau BabelNet.

Leur système a renvoyé le même nombre de réponses que les données qu'ils ont référencées (224 370 occurrences du mot, dont 191 146 occurrences étaient polysémie). Pour mesurer la performance de leur système de désambiguïsation, ils n'ont pas considéré les mots pour lesquels BabelNet ne renvoie qu'un sens candidat. Ils mesurent cette performance en utilisant la précision.

Pour comparer les performances de leur méthode, ils ont choisi de mener des expérimentations sur tous les mots polysémiques du corpus d'évaluation d'une part, et sur un échantillon de mots polysémiques d'autre part.

Les mots de l'ensemble de test sont sélectionnés en fonction de leur degré d'ambiguïté (clair, ambigu ou très ambigu). Ils ont sélectionné quatre mots pour les catégories grammaticales des noms, verbes et adjectifs : nom = {argument, disc, paper, plan}, verbe = {operate, note, add, begin}, adjectif = {black, valid, wet, narrow}. Ils ont un ensemble de 12 mots représentant 1022 occurrences dans SemCor. Ils comparent leurs résultats avec Babelfy (Moro et al. 2014), un système de désambiguïsation qui utilise BabelNet comme base de connaissances.

Ils notent que l'utilisation de voisins distribués non seulement réduit cette complexité mais maintient également la cohérence au niveau de la désambiguïsation. Pour lever l'ambiguïté des noms et des adjectifs, il est toujours préférable d'utiliser des voisins distribués plutôt que des voisins linéaires les plus proches. A plus long terme, **ils** envisagent d'étendre ces recherches en étudiant d'autres similitudes sémantiques au-delà de simples comparaisons d'égalité des traits sémantiques entre les sens pour améliorer les performances de leur système. [27]

5.3. Approche basée sur les arbres sémantiques pour la désambiguïsation lexicale de la langue arabe en utilisant une procédure de vote :

Laroussi Merhbene, Anis zouaghi et Mounir zrigui (2014) présentent une méthode semi-supervisée de désambiguïsation lexicale des mots arabes. Le principal inconvénient de l'arabe

semble être le grand nombre de mots ambigus sortis de leur contexte. La partie supervisée de leur méthode utilise des corpus et des dictionnaires comme ressources pour classer le contexte des mots ambigus selon leur sens. L'étape de regroupement de sens qu'ils emploient est très utile pour atteindre les performances obtenues par leur méthode. En revanche, une représentation sémantique arborescente de chaque sens leur est très pratique. Ils mettent ensuite en correspondance l'arbre sémantique (pour chaque sens) avec l'arbre de la phrase à désambiguïser pour obtenir un graphe acyclique pondéré. Ils définissent une nouvelle métrique de notation (utilisant trois métriques de collocation) pour trouver l'arbre sémantique le plus proche. La partie non supervisée de ce travail est basée sur une procédure de vote qui permet de classer les mesures co-localisées et de sélectionner le sens correct des mots ambigus.

Les méthodes de désambiguïsation de mots semi-supervisées sont une combinaison de méthodes supervisées et non supervisées. Ils ont développé une structure appelée arbre sémantique à partir de méthodes de représentation de clusters telles que les arbres lexicaux et les réseaux (Mihalcea, 2004) et (Navigili et al., 2005). Pour déterminer le sens exact, ils définissent une nouvelle mesure de similarité basée sur un graphe (obtenue en faisant correspondre l'arbre sémantique avec l'arbre de la phrase à désambiguïser) pour trouver l'arbre sémantique le plus proche du sens. Mots à lever l'ambiguïté pour l'arbre généré contenant la phrase originale. Ces derniers peuvent avoir plus d'un sens, c'est pourquoi ils définissent la procédure de vote.

La vérification du sens des mots est l'un des principaux problèmes du travail de désambiguïsation du vocabulaire. Ils définissent une méthode qui génère automatiquement toutes les significations possibles d'un groupe de mots ambigus (mots clés appartenant au paragraphe de mots ambigus) afin qu'il puisse être défini.

Utilisant le corpus en prétraitement, ils ont collecté des phrases contenant les racines des mots à désambiguïser (exp : pour les mots "العين" "Alayn" il faut chercher les racines "عين" "ayn"), puis ils ont éliminé les mots vides qui apparaissent fréquemment dans le corpus et n'affectent pas le sens du mot. Dans leur travail, ils ont utilisé une liste générique de 29985 mots vides. La liste a été compilée par des linguistes arabes et est considérée comme suffisante pour lever l'ambiguïté du sens des mots.

Extraction des racines : Pour extraire les racines des mots arabes, ils ont utilisé l'algorithme sans ressources "Al Shalabi Kanaan et Al serhan" (Al Shalabi et al., 2003), qui permet l'enracinement en attribuant des poids aux lettres qui composent le mot et rang. Regroupement de sens : L'idée du regroupement des senses et de regrouper les phrases extraites du corpus à l'aide des racines des mots appartenant aux gloses, elles utilisent la liste des racines obtenue à l'étape précédente

et l'algorithme de recherche d'approximation des sous-chaînes dans la chaîne (Elloumi, 1998) pour trouver les occurrences possibles des racines.

Dans leur travail, ils ont choisi de représenter du texte (c'est-à-dire des groupes) avec des arbres binaires. Ce choix s'explique par les besoins de leur méthode, la rapidité d'étude des arbres, la compacité de la représentation et la simplicité de l'algorithme de calcul.

La mesure de score proposée (pour mesurer la correspondance entre l'arbre sémantique et l'arbre de la phrase originelle) utilise trois mesures de collocations qui seront classés en utilisant une procédure de vote supervisé.

Les mesures de collocations sont les suivantes :

- Le T-test : $wc_{ij} = T = (\bar{x} - \mu) / \left(\sqrt{\frac{S^2}{N}} \right)$
- Le Khi Carré :

$$\chi^2 = \frac{N \times \left(C_{1,1} \times C_{2,2} - C_{1,2} \times C_{2,1} \right)^2}{\left(C_{1,1} + C_{1,2} \right) \times \left(C_{1,1} + C_{2,1} \right) \times \left(C_{1,2} + C_{2,2} \right) \times \left(C_{2,1} + C_{2,2} \right)}$$

- Information Mutuelle :

$$IM(w_i, w_j) = \log_2 \frac{P(w_i, w_j)}{P(w_i)\, P(w_j)}$$

La mesure de score définit dans ce qui suit nous permet de trouver l'arbre sémantique T_{st} la plus proche à l'arbre de la phrase originelle T_{os} :

$$Score = \sum N_i \in T_{os} \left(\sum N_j \in ST_{S_k} \left(wc_{ij} / ST_{S_k} \left(L(N_j) \right) Nb\left(ST_{S_k} \right) \right) / Nb(T_{os}) \right)$$

Les ressources utilisés dans leur travail, pour la désambiguïsation de la langue arabe ils ont besoin d'un dictionnaire arabe-arabe qui contient les différents sens du mot ambigu, ils utilisent le dictionnaire « Alwassit » (Muṣṭafá et al. 2008) qui est très connu pour la langue arabe et contient les anciens et nouveaux sens. Le corpus utilisé est l'ensemble de plusieurs corpus collectés, Le nombre total de mots dans le corpus est 123, 8554,642 mots.

Leurs résultats indiquent que le χ2 est la mesure de collocation ayant le rang le plus élevé pour la majorité des données testées. Les mots ambigus ayant le nombre de sens le moins élevé donnent les meilleures performances. Ceci s'explique par le fait qu'elles facilitent le choix du sens correct.

Dans cette étude expérimentale ils ont testé des mots arabes ambigus choisis par leur nombre de sens hors de contexte. Les résultats montrent que leur méthode permet d'obtenir un taux de rappel et de précision élevé (83%). [28]

5.4. Approche de désambiguïsation lexicale à base de connaissances par la sélection distributionnelle et traits sémantiques :

Mokhtar Boumedyen Billami propose approche d'analyse distributionnelle et utilisé dans la tâche de la désambiguïsation lexicale. Utiliser une méta-heuristique d'optimisation combinatoire qui consiste à choisir les voisins les plus proches par sélection distributionnelle autour du mot à désambiguïse.

La clé de sa démarche de démystification est de noter les voisins de chaque mot polysémantique dans le texte : au lieu de comparer chaque sens du mot à illustrer avec tous les sens de tous les mots qui apparaissent dans le texte, il fait une comparaison uniquement avec les sens des voisins sélectionnés par similarité distributionnelle. D'une part, ces voisins fournissent souvent des indices sur le sens le plus probable du mot dans le texte. D'une part, cela lui permet de réduire le temps d'exécution de l'algorithme et de ne pas perdre de cohérence dans l'épellation de tous les mots du texte.

corpus de travail: pour le corpus de donnee à disposition un ensemble de trois corpus de différents genres :

1- corpus est une collection de l'agence française de presse (*French press agency*).

2- deuxième corpus est une collection d'articles d'un journal local français (*l'EST Républicain*).

3 - corpus est une collection d'articles issue de la ressource encyclopédique libre, *Wikipédia*

Corpus d'évaluation : Il travaille sur deux corpus différents, corpus IREST contenant 10 textes et un corpus brut contenant 20 textes pour un total de 30 textes. Nous avons 6 235 occurrences de mots (4 139 occurrences de mots pleins) et une moyenne de 208 occurrences (138 occurrences de mots pleins) par texte.

Il a utilisé La version 2.5.1 de BabelNet a été utilisée pour créer le système de désambiguïsation et de détection d'entités nommées Babelfy. Babelfy réalise de bonnes performances grâce à la structure de BabelNet, qui permet l'intégration d'entités de sens de dictionnaire et d'encyclopédie dans un réseau sémantique. Il a évalué ses expériences en utilisant la version 2.5.1 afin de comparer ses résultats avec ceux retournés par Babelfy.

Similarité distributionnelle : La similarité de distribution est une mesure du degré de cooccurrence entre un mot cible et ses voisins dans des contextes similaires. Par exemple, dans un premier texte les voisins de fleuve peuvent être rivière, eau, affluent. Le sens le plus probable pour *fleuve* est décrit dans BabelNet par trois définitions :

(1) Cours d'eau naturel ;
(2) En hydrographie, une rivière est un cours d'eau qui s'écoule sous l'effet de la gravité et qui se jette dans une autre rivière ou dans un fleuve, contrairement au fleuve qui se jette, lui, selon cette terminologie, dans la mer ou dans l'océan ;
(3) Courant d'eau qui coule d'une altitude élevée à une altitude basse pour arriver dans un lac ou une mer, sauf dans les aires désertiques ou il peut arriver sur rien.
La similarité distributionnelle entre deux motsw_1 etw_2 est définie par la fonction suivante :

$$Sim(w_1, w_2) = \frac{2xI\big(F(w_1) \cap F(w_2)\big)}{I\big(F(w_1)\big) + I\big(F(w_2)\big)}$$

$F(w_1)\ et\ (w_2)$ représentent l'ensemble des traits syntaxiques possédés respectivement par $w_1$et w_2. $(w_1)\cap F(w_2)$ représente l'ensemble des traits syntaxiques communs de w_1 et w_2.

Pour mesurer la similarité sémantique, il a utilisé l'algorithme de Lesk (1986) et ses variantes Algorithme de base de Lesk, Variante de Lesk, Algorithme de Lesk étendu. Il a basé son approche sur la mesure de similarité distributionnelle de Lin (1998) pour déterminer un score entre le mot cible et l'ensemble des mots du paragraphe qui appartiennent à la même catégorie grammaticale du mot cible.

Pour mesurer les performances des différentes méthodes de désambiguïsation, il a utilisé le taux d'exactitude (*accuracy*), L'évaluation de sa méthode est effectuée sur des données dont la couverture des sens par BabelNet est de 100%. Ce taux d'exactitude est calculé pour chaque mot du jeu de test et pour chaque méthode de désambiguïsation testée. Il présente le rapport entre le nombre d'occurrences correctement désambiguïsées et le nombre total d'occurrences d'un mot. Leur évaluation porte d'une part sur le niveau d'ambiguïté des mots polysémiques, d'autre part, sur la mesure distributionnelle utilisée pour choisir les *k-plus proches voisins* (*k*-PPV).

Après ses expériences, il a été conclu que l'algorithme Lesk étendu retourne le meilleur résultat par rapport à la baselesk et Babelfy pour les noms et cela sur toutes les variations utilisées pour obtenir un ensemble des triplets de dépendances syntaxiques. Pour les verbes, quelques sélections aléatoires des triplets de dépendances apportent à leur approche des résultats faibles par rapport à la base lesk et Babelfy. L'utilisation d'une autre mesure de similarité qui ne repose pas sur des

traits sémantiques peut corriger ce problème.

La méthode il a testé et évaluée s'appuie sur une sélection distributionnelle des voisins les plus proches selon le contexte du mot à désambiguïser. Il a adapté l'application d'une approche exhaustive pour se comparer avec *k*-PPV. Cette approche adaptée repose sur l'utilisation de mesures de similarité à base de traits sémantiques. Le contexte utilisé dans cette expérience correspond à un paragraphe et le corpus d'évaluation appartient à un domaine général et non pas à un domaine de spécialité. Le meilleur taux d'exactitude retourné pour les noms est de 90,91% contre 60,71% pour les verbes. La meilleure combinaison retourne 77,78% (k=5 et 30%V1 de l'ensemble des triplets de dépendances syntaxiques). [29]

5.5. Approche de désambiguïsation morpho_lexicale évaluée sur l'analyseur morphologique Alkhalil :

K.Z Bousmaha, S. Charef_Abdoun1, L. Hadrich_Belguith, M.K Rahmouni travaillé sur une approche de désambigüisation se fonde sur le choix de la bonne diacritisation du mot du texte du fait que le mot arabe en accepte plusieurs. Bien diacritiser revient à choisir en contexte la bonne diacrité d'un mot afin d'en déterminer le sens et la fonction. La problématique est double Comment restituer les diacritiques potentielles de chacun des mots d'un texte analysé morphologiquement, alors que plusieurs catégories grammaticales peuvent être affectées à un mot et Comment choisir le bon schème diacritique parmi tout un ensemble proposé pour une même catégorie grammaticale attribuée à ce mot ? La tâche de désambiguïsation semble alors difficile.
Leur choix s'est penché en premier lieu sur une méthode multicritère d'aide à la décision à base de TOPSIS (Technique for Order by Similarity to Ideal Solution) du fait de sa robustesse et de son fondement mathématique**,** son fondement consiste à choisir une solution qui se rapproche le plus de la solution idéale, en se basant sur la relation de dominance qui résulte de la distance par rapport à la solution idéale (la meilleure sur tous les critères) et de s'éloigner le plus possible de la pire solution (qui dégrade tous les critères). Il s'agit de réduire le nombre de scénarios1 de désambiguïsation, et de classer les scénarios efficaces selon leurs scores globaux calculés.

Leur approche se compose de deux étapes :

La premiére commence par une analyse linguistique**.** Après segmentation en phrases**,** une analyse morpho_syntaxique est effectuée dont l'objectif est d'associer à chaque unité lexicale sa catégorie grammaticale (nom, verbe, adjectif...). Le principal intérêt de cet étiquetage est qu'il permet d'opérer un premier traitement de désambiguïsation des mots. L'étiqueteur utilisé peut ainsi associer plusieurs étiquettes (catégories) à chaque unité lexicale (surtout pour un mot ambigu).

La deuxième étape consiste en une application de règles de grammaire implémentées sous forme d'un ensemble de réseaux de transitions augmentés (ATN) sur le texte préalablement étiqueté. Il doit

par conséquent choisir parmi toutes les catégories affectées précédemment à une unité lexicale celle qui correspond au mot dans la phrase considérée, en s'appuyant sur l'ATN correspondant. Une fois le trait grammatical associé, une deuxième désambigüisation est ainsi faite, les (ou quelques) diacritiques les plus évidentes de ce mot sont trouvées et l'ensemble des schèmes candidats associé au mot est ainsi réduit.

En vue d'améliorer les performances du système en termes de qualité de solutions et de temps d'exécution, et afin de prendre en considération ces problèmes d'ambiguïtés au cours de l'analyse de la langue, il n'est plus à démontrer qu'une architecture multi-agent de ses différents modules est à envisager (beaucoup d'analyseurs pour le TALA adoptent cette stratégie : mouhalil, maspar, etc.

Ils ont choisi Alkhalil Morpho Sys, un analyseur morphologique Ligne de texte arabe standard. Alkhalil peut gérer du texte sans signes diacritiques, il peut donc Gère le texte partiellement ou entièrement diacritique. Nous avons choisi Alkhalil Morpho Sys) car il peut être considéré comme le meilleur système morphologique Arabe. En fait, Alkhalil a pris la première place sur 13 systèmes Formes arabes à travers le monde, dans un concours organisé par la Ligue arabe Éducation, Culture et Sciences (ALECSO).

Architecture de L'analyseur D_Alkhalil

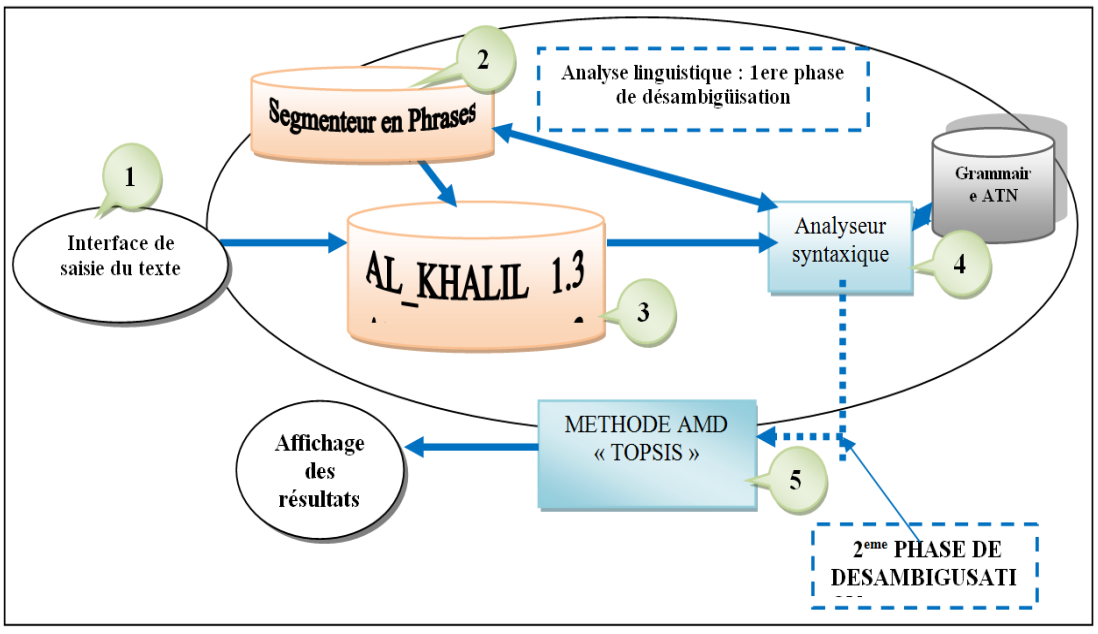

Figure 9: Architecture D_Alkhalil

Lors de la saisie de texte à Alkhalil, Alkhalil analyse les données reçues (analyse lexicale, analyse syntaxe….). L'analyse lexicale-syntaxique se déroule en cinq étapes : prétraitement , Segmentation, analyse de la racine, dépistage des résultats , affichage des résultats de l'analyseur morphosyntaxique après cela, il est passé à d'autres étapes, qui est l'analyse syntaxique , Une autre phase de désambiguïsation: cette phase permet de réduire le nombre d'interprétations issues des premières analyses grâce à la démarche multicritères à base de la méthode TOPSIS.

Pour mieux comprendre le principe de la méthode, ils l'ont illustré par une extension.

Exemple :

Ph = «ذهب الطفل إلى البستان», Analysons le mot « ذهب »

Étape 1 : détermination des scénarios :

E= {Verbes : فَعَلَ, فَعِلَ, فَعُلَ, فَعِلٌ, فُعِلَ}

Étape 2 : Application des critères sur les scénarios :

Scénarios :{فَعَل111, فَعِلَ 101, فَعُلَ 101, فَعِلٌ 100 ,001 فُعِلَ}

Étape 3et 4 : Appliquer de la fonction d'évaluation et Générer la matrice d'évaluation Exemple :

Étape 6 : Détermination de la solution idéale et la solution anti idéale, en adoptant la formule de Topsi.

ذهب	ذَهَبَ	فعل ماض مبني للمعلوم	فَعَلَ	ثلاثي مجرد مسند إلى الغائب(هو) متعد ولازم

Figure 10: Résultat après désambiguïsation par l'analyseur D_Alkhalil

Après avoir fait ces expériences, le groupe a conclu que : les résultats de données. Les exemples pris démontrent une désambigüisation d'Al_khalil de plus de 85% pour certains cas.

L'ambiguïté n'a pas été poussée à l'extrême et les résultats obtenus ne sont pas Cent pour cent (100%) fiable car les principaux bugs sont dus à :

- Données non incluses dans la base de données d'Alkhalil.
- Pour 22,5% des phrases analysées, leur échec d'analyse est principalement dû à Le fait que leur architecture ne soit pas couverte par leurs règles (c'est le cas avant Exemple de phrases longues ou de phrases imparfaites et/ou d'ellipses non Reconnu) ou encore pour hacher l'échec en chaînes, l'échec de reconnaître les propriétés morpho_syntaxiques de certains mots, etc.).
- Le besoin d'une autre clarification sémantique que nous envisageons Pour l'appliquer à l'aide de l'ontologie arabe WordNet, qui est précise dans Couvrir le sens des termes.

D_Alkhalil en est encore à ses balbutiements, car des extensions sont en cours :

− Enrichissement de la base de données Al_khalil à considérer mot inconnu.

− Élargissement de la catégorie grammaticale d'Alkhalil.

− Étendre la base de règles de grammaire de l'analyseur.

- Implémentation à l'aide de méthodes de désambiguïsation sémantique réseau arabe. [30]

- **Avantages de chaque approche :**

Table 1: avantages de chaque approche

Approches	Avantages de chaque approche
Représentation vectorielle de sens pour la désambiguïsation lexicale à base de connaissances	représenterles significations dudictionnairesousforme vectorielle. prendlestermesutilisésdansladéfinition, les projettent dansunespace vectoriel, puis additionnent les vecteurs résultants, enles pondérant enfonctiondeleurpartiedudiscoursetdeleurfréquence
Approches d'analyse distributionnelle pour améliorer la désambiguïsation sémantique	réduirelacomplexité exponentielle en sélectionnant lesvoisinsdedistributionlesplus proches sansperdredecohérence au niveaudela désambiguïsation. Déterminer ladistributiondesplus proches voisinspourchaque polysémie dansle texte.
Approche basée sur les arbres sémantiques pour la désambiguïsation lexicale de la langue arabe en utilisant une procédure de vote	présenter une méthode semi-supervisée de désambiguïsation lexicale des mots arabes. La partie supervisée utilise des corpus et des dictionnaires comme ressources pour classer le contexte des mots ambigus selon leur sens, et la partie non supervisée de basée sur une procédure de vote qui permet de classer les mesures colocalisées et de sélectionner le sens correct des mots ambigus.
Approche de désambiguïsation lexicale à base de connaissances par la sélection distributionnelle et traits sémantiques	Utiliser une méta-heuristique d'optimisation combinatoire qui consiste à choisir les voisins les plus proches par sélection distributionnelle autour du mot à désambiguïse.
Approche de désambiguïsation	Dépenddu choix de la bonne diacritisation du mot

morpho_lexicale évaluée sur l'analyseur morphologique Alkhalil	du texte du fait que le mot arabe en accepte plusieurs. Adopter (Alkhalil Morpho Sys) car il peut être considéré comme le meilleur système morphologique arabe.

6. La désambiguïsation dans les applications de TALN :

6.1.Traduction automatique :

La traduction automatique est la première des applications ayant considéré la désambiguïsation sémantique comme une tâche intermédiaire fondamentale. Il s'agit donc d'un domaine de recherche par excellence où il est crucial de lever l'ambiguïté sémantique des mots afin d'aboutir à des traductions correctes. Par exemple, la traduction en anglais du mot français *glacial* est *icy* ou *bitter* selon s'il s'agit du froid ou d'une personne blessée (ou en colère). [31]

Traduction automatique ou (Machin Translation MT) est l'application la plus évidente de WSD (Word Sens Desamb iguation). En MT, le choix lexical des mots qui ont des traductions distinctes pour différents sens est effectué par WSD. Les sens en MT sont représentés sous forme de mots dans la langue cible. La plupart des systèmes de traduction automatique n'utilisent pas de module WSD explicite. [32]

6.2.Recherche d'information (RI) :

Recherche d'information (RI) peut être défini comme un logiciel qui traite de l'organisation, du stockage, de la récupération et de l'évaluation d'informations à partir de référentiels de documents, en particulier d'informations textuelles. Le système aide essentiellement les utilisateurs à trouver les informations dont ils ont besoin, mais il ne renvoie pas explicitement les réponses aux questions. WSD est utilisé pour résoudre les ambiguïtés des requêtes fournies au système IR. Comme pour MT, les systèmes IR actuels n'utilisent pas explicitement le module WSD et ils s'appuient sur le concept selon lequel l'utilisateur saisirait suffisamment de contexte dans la requête pour ne récupérer que les documents pertinents. [31]

6.3.Traitement de la parole :

La phonétisation correcte des mots en synthèse de la parole demande une tâche de désambiguïsation. Cette tâche est également utilisée en reconnaissance de la parole pour la segmentation des mots et pour la discrimination d'homophones. Ces derniers représentent des mots qui se prononcent de manière identique mais dont le sens est différent. [31]

6.4.Exploration de texte et extraction d'informations (IE) :

Dans la plupart des applications, WSD est nécessaire pour effectuer une analyse précise du texte. Par exemple, WSD aide système de collecte intelligent pour signaler les mots corrects. Par exemple, un système médical intelligent pourrait avoir besoin de signaler les « drogues illégales » plutôt que les « médicaments ». [32]

6.5.Lexicographie :

La désambiguïsation sémantique et la lexicographie (*i.e.*, la réalisation de dictionnaires) peuvent certainement bénéficier l'une de l'autre. D'une part, la désambiguïsation sémantique peut aider à fournir des groupements de sens empiriques et indices statistiques contextuels pour des nouveaux sens ou des sens existants, comme elle peut aider à créer de nouveaux dictionnaires plus lisibles. D'autre part, un lexicographe peut fournir de meilleurs inventaires de sens et des corpus annotés sémantiquement dont le bénéfice sera pour l'utilisation des méthodes de désambiguïsation sémantique. [31]

WSD et lexicographie peuvent fonctionner ensemble en boucle car la lexicographie moderne est basée sur des corpus. Avec la lexicographie, WSD fournit des regroupements de sens empiriques approximatifs ainsi que des indicateurs contextuels de sens statistiquement significatifs. [32]

6.6.Récupération de l'information :

Le paradigme dominant en RI est basé sur les représentations en sac de mots : un morceau de texte est caractérisé comme une collection non ordonnée de termes, et l'évaluation de la pertinence d'un document en réponse à une requête dépend principalement des termes qu'ils ont en commun. Intuitivement, les termes sont les mots eux-mêmes. En pratique, les mots communs non informatifs sont exclus en tant que termes, et plusieurs formes de mots sont mappées à une forme unique via la racine, par exemple, connecter, connecter, connecter et connexion seraient tous issus de la connexion. Par conséquent, une requête sur la connexion de mon appareil photo et un document contenant la connexion d'un appareil photo numérique auraient en commun les termes connecter et appareil photo. Trois raisons à cela ont été largement notées. Premièrement, si les requêtes sont courtes, le contexte disponible pour la désambiguïsation basée sur le contexte des termes de requête est extrêmement limité, ce qui rend le WSD difficile. Deuxièmement, même pour les mots à plusieurs sens, le sens le plus fréquent domine souvent fortement la distribution de fréquence de la collection de textes considérée ; dans un tel les cas utilisant le mot lui-même sont susceptibles d'être tout aussi bons que la désambiguïsation correcte. Troisièmement, la plupart des modèles de récupération de documents présentent une tendance à la désambiguïsation implicite des requêtes multi-mots, ce qui aide l'IR à sac de mots à bien fonctionner même en l'absence de sens explicite des mots, en particulier pour les requêtes plus longues. [33]

6.7.Analyse thématique et **Analyse grammaticale** :

Les thèmes sont identifiés en fonction de la distribution des mots, mais nous n'incluons que les mots du sens pertinent.

Dans le balisage POS ou l'analyse syntaxique, WSD est utile. Dans la phrase française «L'étagère pliée sous les livres», livres fait référence à des «livres» et non à des «livres». [33]

6.8.Substitution lexicale :

La substitution lexicale est une tâche qui, ces dernières années, a reçu un intérêt majeur au sein de la communauté du TALN. Le principe consiste à remplacer un mot-cible par un substitut potentiel tout en gardant le même sens du mot-cible par rapport au contexte dans lequel il apparaît.

La substitution lexicale reflète non seulement les capacités des systèmes de désambiguïsation sémantique à choisir le bon sens, mais peut également être utilisée pour comparer les ressources lexicales. Elle a le potentiel d'être elle même bénéfique pour d'autres applications (par exemple, dans le cadre d'une simplification automatique de textes). [31]

7. Conclusion :

L'ambiguïté est encore à ce jour l'un des principaux problèmes en TAL, malgré les différentes approches qui ont été proposées. On a présenté durant ce chapitre les différentes définitions et les types d'ambiguïté , Comme nous avons abordé La désambiguïsation, elle consiste à déterminer le sens le plus approprié pour chaque mot dans le texte à son emplacement prédéfini. nous avant présenté quelques approches de la désambiguïsation. Nous avons conclu en énumérant les différents domaines et applications de la TALN qui ont été adoptés .

Chapitre 03 : les mesures de similarité

1. Introduction

En informatique, les mesures de similarité sont des fonctions utilisées dans plusieurs domaines, notamment le traitement automatique du langage (TAL), la recherche d'informations, la traduction automatique, le résumé de texte, etc. Les mesures de similarité jouent un rôle important, notamment dans le processus de la désambiguïsation des termes. L'objectif principal des mesures de similarité est d'estimer la ressemblance entre les concepts.

La plupart des mesures de similarité sont basées sur des aspects statistiques et ne tiennent pas compte des relations sémantiques qui existent entre les mots de la langue. Ces dernières années, plusieurs travaux de recherche ont proposé de nouvelles mesures de similarité basées sur les relations sémantiques entre les mots du langage. [34]

Dans ce chapitre nous aborderons la définition du concept de mesure de similarité et ses usages, ainsi que la similarité statistique, la similarité sémantique et ses méthodes.

2. Définition d'une mesure de similarité :

Une mesure de similarité S est une fonction $X * X \rightarrow R$ qui satisfait les propriétés suivantes :

- Positivité : $\forall x, y \in X, S(x, y) \geq 0$.
- Symétrie : $\forall x, y \in X, S(x, y) = S(y, x)$.
- Maximalité : $\forall x, y \in X, S(x, x) \geq S(x, y)$.

D'autres attributs peuvent être requis, tels que la normalisation, qui requiert la valeur Appartient à l'intervalle [0, 1].

Les mesures non standard peuvent être converties normaliser pour obtenir leurs versions normalisées. Ensuite, nous considérons le cadre Mesures standardisées. [35]

3. Utilisation de mesures de similarité :

La mesure de la similarité sémantique entre les concepts est un problème important dans le domaine du crawling web et du texte mining. Exploration de texte nécessitant une correspondance de contenu sémantique.

- De nombreuses applications du monde réel utilisent des mesures de similarité pour Les objets sont interdépendants. Nous pouvons utiliser ces mesures dans des applications impliquant Vision par ordinateur et traitement du langage naturel, tels que Find and Match fichiers similaires. Un cas d'utilisation important pour les entreprises consiste à faire correspondre Faire correspondre les CV aux descriptions de poste fait gagner beaucoup de temps aux recruteurs (l'embauche de personnel).Un autre cas d'utilisation important est l'utilisation de clusters K Means pour

segmenter différents clients pour une campagne marketing. Une campagne marketing utilisant l'algorithme de clustering K-means, qui utilise également une mesure de similarité.

- La similarité est au cœur de plusieurs emplois dans différents domaines, tels que l'analyse de données, Raisonnement par cas, reconnaissance de formes, résolution de problèmes, apprentissage, transfert...
- Exprimer la similitude sémantique de la connexion entre deux concepts est la capacité abstraite des êtres humains, et la machine n'a pas la capacité de l'expliquer. De toute évidence, les concepts de "stylo" et de "papier" sont plus liés que les concepts de "température" et de "chaise". Cet état de fait est difficile à formaliser sans recourir à des ressources sémantiques : les ontologies permettent de mettre en évidence des liens entre concepts (synonymes, antonymes, etc.)

3.1. Similarité dans l'analyse de données :

De nombreuses techniques de science des données sont basées sur la mesure des similitudes et des dissemblances entre objet. Par exemple, K-Nearest-Neighbors utilise la similarité pour classer de nouveaux objets de données. Dans l'apprentissage non supervisé, K-Means est une méthode de clustering qui utilise la distance euclidienne pour calculer la distance entre les centres de gravité du cluster et le point de données qui lui est attribué. Le moteur de recommandation utilise une méthode de filtrage collaboratif basée sur le quartier Identifiez les voisins d'un individu en fonction de leur similarité/dissemblance avec d'autres utilisateurs.

3.2. Similarité dans la reconnaissance des formes :

Les motifs sont des éléments de l'espace de modèles ou des classes hypothétiques, et les données fournissent "Information" lequel de ces modèles doit être utilisé pour expliquer les données, Cartographie la relation entre les données et le schéma est construite par des algorithmes de raisonnement, notamment à travers le coût minimisez le processus. La volatilité des données limite souvent notre précision permet l'identification unique d'un modèle unique en tant qu'interprétation des données

3.3. Similarité dans le raisonnement basé sur les cas :

La similarité est un concept central du raisonnement par cas (CBR : Case - based reasoning), puisque la construction de règles de cas, la récupération de cas et même l'adaptation à une situation utilisent toutes une logique basée sur l'analogie ou la similarité. Tout cela est un peu déroutant en utilisant échelles de similarité et de similarité et échelles de similarité dans CBR, en particulier dans les systèmes CBR dépendants du domaine.

Le raisonnement par cas (CBR) est l'un des paradigmes émergents pour la conception de systèmes intelligents. L'extraction de cas similaires est une étape clé dans RBC, et la mesure de

similarité joue un rôle très important dans la découverte de cas. Parfois, un système RBC est appelé un système de recherche de similarité, et sa caractéristique la plus importante est l'efficacité d'une mesure de similarité utilisée pour quantifier le degré de similarité entre une paire de cas. [36]

4. Les mesures de similarités statiques :

Parmi les modèles de la CT (Catégorisation de Texte) qui utilisent les mesures statistiques, on trouve le modèle booléen qui se base sur la présence / absence de terme dans un document, et le modèle vectoriel qui utilise les fonctions de similarité, on peut distingue les meures suivants :

4.1. Le Produit scalaire : se calcule par la formule suivante :

$$\boldsymbol{\Sigma_{t\epsilon T} W_{t,A} * W_{t,B}}$$

Tel que :

T : l'ensemble des attributs.

A et B : deux documents (A le document à classé et B le document classé).

$W_{t,A}$: Le poids de terme dans le document A

$W_{t,B}$: Le poids de terme dans le document B

L'inconvénient de produit scalaire c'est que les résultats obtenus sont non normalisés.

4.2. La mesure de Cosinus : après la négativité que les chercheurs sont trouvée dans la méthode du produit scalaire, ils sont proposent la mesure de cosinus qui donne des résultats normalisés qui est entre 0 et 1, elle est définie par la formule suivante :

$$\boldsymbol{\Sigma_{t\epsilon T} \frac{p_t(a).p_t(b)}{\sqrt{\Sigma_{t\epsilon T} p_t(a)^2 . \Sigma_{t\epsilon T} p_t(b)^2}}}$$

Avec :

T : l'ensemble des attributs.

$p_t(a)$: Le poids de terme t dans le document a.

$p_t(b)$: Le poids de terme t dans le document b.

4.3. La mesure de Dice :il été proposées dns la littérature, il est représetépar la formule suivant :

$$\textbf{Dice (A, B)} = \frac{2N_c}{N1+N2}$$

Où :

A et B : deux documents.

N_c : Le nombre de termes communs entre A et B.

N1 (resp. N2) : est le nombre de termes dans le document A et (resp. B).

4.4. La mesure de JACCARD : définit par :

$$\textbf{JACCARD (A, B)} = \frac{2 * \Sigma_{i=1}^{t} A_i * B_j}{Min\left(\Sigma_{i=1}^{t} A_i^2 \Sigma_{i=1}^{t} B_j^2\right)}$$

Tel que :

A et B deux documents (A le document a classé) B le document classé. [34]

5. Les mesures de similarités sémantiques.

La similarité sémantique est dans un ensemble de documents ou conditions. L'idée de la distance entre les documents et les termes est basée sur similitude dans leur signification ou leur contenu sémantique, plutôt que La similarité peut être estimée à partir de leur représentation syntaxique (par par exemple leur format de chaîne). [37]

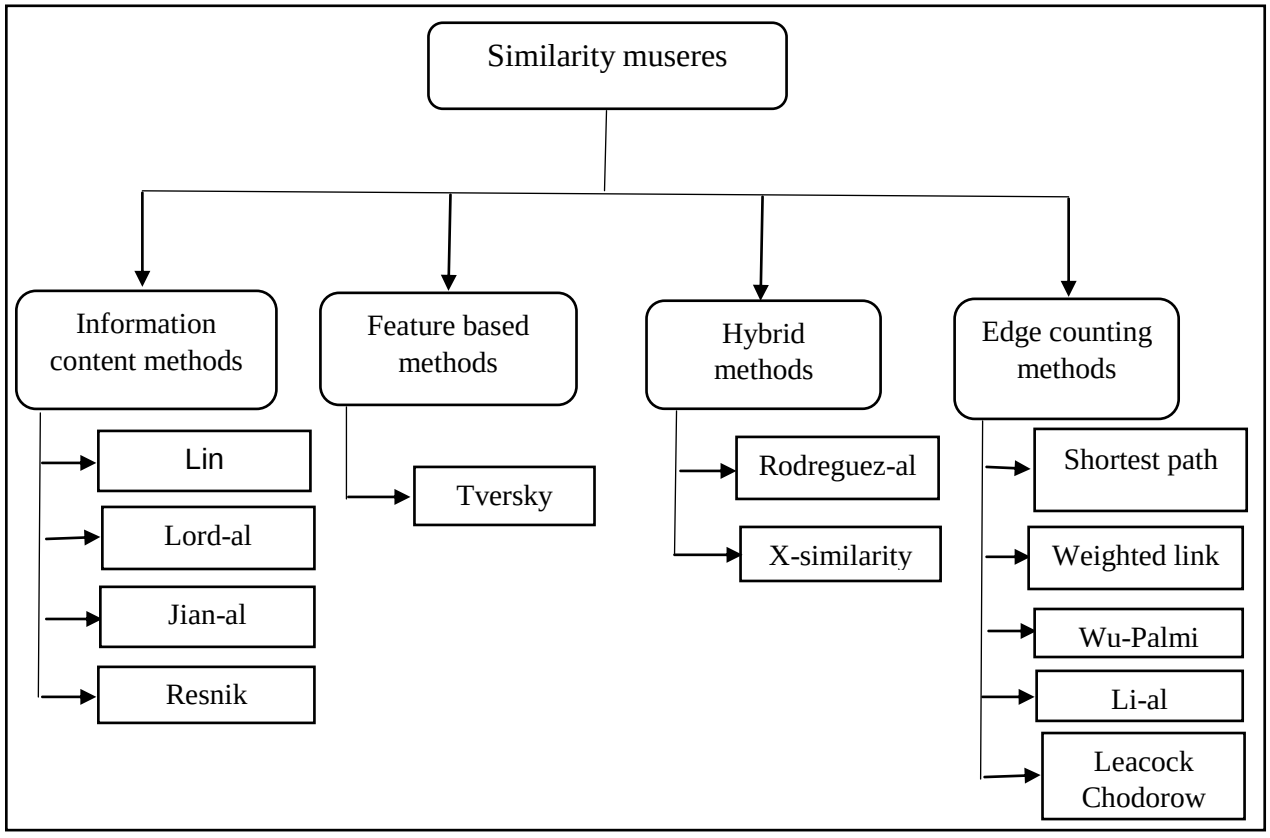

Figure 11: Taxonomie des mesures de similarité sémantiques. *[48]*

5.1. Méthodes de contenu de l'information (information content methods) :

La méthode vise à compléter la structure taxonomique d'une ontologie en utilisant la distribution des informations conceptuelles évaluées dans l'ensemble d'entrée. Utilisant le concept de contenu

informationnel (IC), il est calculé en associant la probabilité d'apparition à chaque concept de la taxonomie, en fonction de leur présence dans le groupe Grant.

IC d'un terme A est calculé selon le log négatif de sa probabilité de présence. De cette manière, les mots peu fréquents sont considérés plus informatifs que les communs.

$$\boldsymbol{IC(A) = -\log\big(p(A)\big)}$$

Parmi les mesures les plus connus de ce méthods : La mesure de Resnik, la mesure de Lin et la mesure de Jiang. [37]

5.1.1. La mesure de Resnik :

Resnik(1995) a été le premier à fusionner l'utilisation d'ontologies avec des corpus. Il définit la similarité sémantique entre deux concepts C1 et C2 par la quantité d'informations qu'ils partagent. Cette information partagée est égale au contenu informationnel du Least Generalizer (LCS) - le concept le plus spécifique qui inclut les deux concepts dans l'ontologie (le concept commun le plus spécifique). [37]

$$\boldsymbol{Sim_{Res} = IC\big(LCS(C1, C2)\big)}$$

5.1.2. La mesure de Jiang-Conrath :

Jiang et Conrath (1998) ont proposé une autre mesure, considérant cette fois aussi la probabilité conditionnelle d'occurrence de deux sens par rapport au concept commun. Simplifiée, cette mesure Il peut être exprimé comme suit : [38]

$$\boldsymbol{Sim_{JnC} = \frac{1}{Dist(C12), C}}$$

5.1.3. La mesure de Lin :

Lin (1998) ont également proposé une mesure de similarité très similaire, qui revient essentiellement à reformuler la formule de Jiang et Conrath sous forme de ratio : [38]

$$\boldsymbol{Sim_{Lin} = \frac{2IC\big(LCS(C1, C2)\big)}{IC(C1) + IC(C2)}}$$

5.2. Méthodes basées sur les fonctionnalités (feature based methods) :

Les mesures basées sur les caractéristiques sont indépendantes de Taxonomie et généralisation des concepts, et tentatives d'utilisation des propriétés ontologiques pour obtenir des valeurs de

similarité. Ontologie pour obtenir la valeur de similarité. Plus deux concepts ont de caractéristiques communes et moins de caractéristiques non communes, plus les concepts sont similaires. Les caractéristiques entre une sous-classe et sa superclasse contribuent davantage à l'évaluation de la similarité que les caractéristiques de la classe inverse. Rapport d'évaluation de similarité dans la direction opposée.

- **Similarité de Tversky :**

$$\boldsymbol{Sim_{Tver}} = \frac{|\boldsymbol{C1} \cap \boldsymbol{C2}|}{|\boldsymbol{C1} \cap \boldsymbol{C2}| + \boldsymbol{k}|\boldsymbol{C1}\backslash\boldsymbol{C2}| + (\boldsymbol{k}-\boldsymbol{1})|\boldsymbol{C2}\backslash\boldsymbol{C1}|}$$

Où :

k est réglable, et k ϵ [0,1]. [36]

5.3. Méthodes hybrides (hybrid methods) :

Une mesure de similarité hybride combine plusieurs mesures de similarité pour produire des scores de similarité plus performants que les scores des mesures individuelles d'entrée que les scores des mesures individuelle d'entrées.

Jiang et Conrath (1998) combinent la méthode de la longueur de chemin et le contenu informationnel (CI) introduit par Resnik :

$$\boldsymbol{Dist}(\boldsymbol{C1}, \boldsymbol{C2}) = \big(\boldsymbol{IC}(\boldsymbol{C1}) + \boldsymbol{IC}(\boldsymbol{C2})\big) - \boldsymbol{2} * \Big(\boldsymbol{IC}\big(\boldsymbol{LCS}(\boldsymbol{C1}, \boldsymbol{C2})\big)\Big)$$

$$\boldsymbol{Sim_{JnC}} = \frac{\boldsymbol{1}}{\boldsymbol{Dist}(\boldsymbol{C1}, \boldsymbol{C2})}$$

Contrairement à la similarité de Jiang et à d'autres méthodes qui utilisent des métriques, il existe des résultats expérimentaux qui prouvent que la similarité n'est pas une pure métrique de distance, et qu'elle ne vérifie pas non plus les propriétés de la distance. Même les méthodes hybrides qui combinent plusieurs sources d'information et peuvent discriminer entre différentes paires de concepts nécessitent un ou plusieurs paramètres, qui doivent être alternés.

Plusieurs paramètres sont nécessaires et il faut tourner également. Bien qu'il n'existe pas de modèle standard pour évaluer les mesures computationnelles de la similarité sémantique (Meng et al 2013), d'autres approches, basées sur les sens communs des concepts, fournissent une nouvelle similarité de sens (Djeddai et al 2012 ; Kalmukov, 2012 ; Moussiades et Vakali, 2009**).**

Prenant deux concepts C1, C2 avec les synsets correspondants Syn(C1), Syn(C2), l'ensemble de leurs sens communs est défini par :

SCom = Syn (C1) ∩ Syn (C2).

SGen = Syn (C1) ∪ Syn (C2).

La mesure de similarité est donnée par l'équation :

$$\boldsymbol{Sim(C1, C2) = 1 - \frac{|SGen| - |SCom|}{|SGen|} = \frac{|SCom|}{|SGen|}}$$

En général, un modèle de mélange se compose de concepts, d'attributs et relation. Les concepts sont modélisés comme des nœuds dans un réseau sémantique. Ils sont décrits par des attributs ou des fonctionnalités dans un ensemble de fonctionnalités. Nous avons remarqué que leur complexité de calcul ne fonctionnait pas bien quand il n'y a pas d'ensemble complet de fonctionnalités. [39]

5.4. Méthodes de comptage des arcs (Edge counting methods) :

Les mesures de similarité basées sur les arcs ont pour principe de compter le nombre d'arcs séparant deux concepts dans une taxonomie.

Dans la figure suivant : C1 et C2 sont deux concepts, qui ont comme plus petit ancêtre C3, et N1, N2 et N3 représentent respectivement le nombre d'arcs entre c3 et c1, c3 et c2 et c3 et la racine. [40]

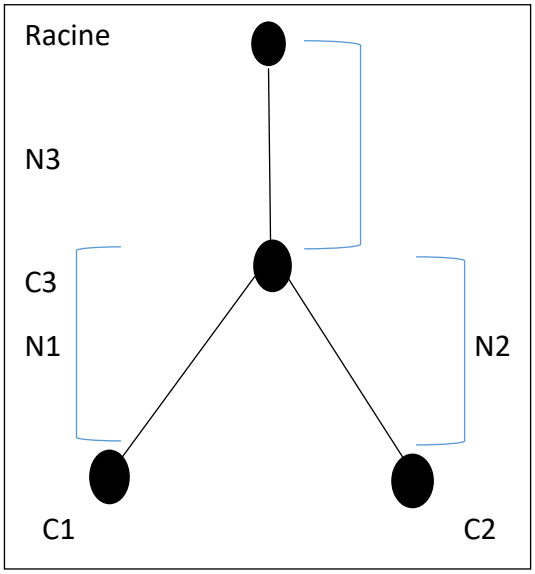

Figure 12: Exemple de taxonomie pour les mesures de similarite basées sur les arcs. *[40]*

5.4.1. *La mesure de Rada* :

Les mesures de similarite sémantique basées sur les arcs ont été introduites par (Rada et al, 1989). Elles ont été définies en fonction de la distance qui sépare deux concepts. La mesure est donnée par l'expression : [40]

$$\boldsymbol{Sim_{Rad}dist(c1, c2) = \frac{1}{1 + dist(C1, C2)} = \frac{1}{1 + N1 + N2}}$$

5.4.2. La mesure de Wu et Palmer :

En 1994, Wu et Palmer proposent une amélioration de la mesure de Rada, en prenant en compte la distance entre l'ancêtre commun le plus spécifique. Originellement leur mesure est destinée originellement à être utilisée avec une taxonomie de verbes, cependant n'importe quelle taxonomie de concepts peut être utilisée. C'est pourquoi en conjonction de WordNet, qui propose des relations taxonomiques pour les verbes et les noms, il est possible de calculer la similarité à la fois entre des verbes et entre des noms, mais pas pour les adverbes où les adjectifs. La mesure s'exprime comme : [19]

$$\boldsymbol{Sim_{WuP} = \frac{2.N3}{N1 + N2 + 2.N3}}$$

5.4.3. La mesure de Leackock et Chodorow :

Cette mesure est basée sur la longueur du plus court chemin entre deux sens. Les auteurs ont limité leur attention à des liens hiérarchiques «is- a » ainsi que la longueur du chemin par la profondeur globale P de la taxinomie. La formule est définie par :

$$\boldsymbol{Sim(X, Y) = -log\left(\frac{cd(x, y)}{2 * M}\right)}$$

Donc :

M : La longueur du chemin le plus long qui sépare le concept racine, de l'ontologie, du concept le plus en bas(les arcs).

***cd* (*x, y*) :** La longueur du chemin le plus court qui sépare X de Y(les nœuds). Cette mesure n'est pas complète car elle ne prend en considération que les hyperonymes et les hyponymes. [34]

6. Comparaison des différentes méthodes de mesure de similarité :

Table 2: Comparaison des différentes méthodes de mesure de similarité. *[36]*

Catégorie	Principe	Mesure	Caractéristiques	Avantages	Désavantages
Basée sur la longueur du chemin (nombre d'arcs)	En fonction de la longueur du chemin entre les concepts et la position des concepts dans l'hiérarchie	Shortest path	Shortest path	simple	Deux paire avec des longueur de chemins les plus courts auront la même similarité
		W&P	Longueur des chemins des concepts à partir de la racine et du subsumant le plus spécifique (LCS), Least Common Subsumer	simple	Deux paires avec la même profondeur du LCS et des longueurs de chemins égales auront le même similarité
		L&C	nombre d'arcs et fonction logarithmique (lissage)	simple	Deux paires avec la même longueur du plus court chemin auront la même similarité
		Li	Fonction non linéaire de la	simple	Deux paires avec le même

			longueur du plus court chemin et la profondeur du LCS		LCS et des longueurs de chemins égales auront le même similarité
IC based	Contenu informative élevé Plus deux concepts partagent un contenu informatif commun, plus ils seront similaire	Resnik	Le contenu informatif du LCS	simple	Deux paires avec le même LCS auront la même similarité
		Lin	IC, Le contenu informatif des concepts et du concept LCS	considère le IC du concept LCS des deux concepts	Deux paires ayant le même IC des concepts comparés et celui de leurs LCS auront la même similarité
		Jiang	IC, Le contenu informatif des concepts et du concept LCS	Considère le IC des concepts	Deux paires ayant le même IC des concepts comparés et celui de leurs LCS auront la même similarité

Méthodes basées caractéristiques	Concepts ayant en commun plus de caractéristiques et peu de caractéristiques différentes sont plus similaires	Tversky	Compare les caractéristiques des concepts	Considère les caractéristiques des concepts	Complexité computationnelle
Méthodes Hybrides	Combine plusieurs paramètres informationnels	Zhou	combines IC et chemins	Distingue mieux les différentes paires de concepts	difficulté de choix des paramètres à combiner et leurs pondérations

7. Evaluations des différentes méthodes par rapport aux jugements humains :

Diverses expérimentations ont été effectuées sur les méthodes de similarités utilisant divers Data sets. Une expérimentation est celle de [41] rapporté dans [42], nous l'avons choisie parce qu'elle a été confrontée et testée par rapport aux jugements humains (ensemble de 30 étudiants), pour calculer la corrélation entre les résultats obtenus par ces mesures et les jugements humains, les résultats obtenus sont résumés dans la table2 ci-dessous :

Table 3: Evaluation des méthodes : Edge Counting, Information Content, Feature-based et Hybride des mesures de similarité appliqués à WordNet. *[42]*

Evaluation des méthodes : Longueur de chemin, contenu informatif, basées caractéristiques et méthodes hybrides de calcul des mesures de similarités appliquées sur WordNet.	**Types de Méthode**	**Corrélation**
Rada & al	Edge Counting	0.59
Wu & Palmer	Edge Counting	0.74
Li	Edge Counting	0.82
Leacock & chodorow	Edge Counting	0.82
Richardson	Edge Counting	0.63
Resnik	Information Content	0.79
Lin	Information Content	0.82
Lord	Information Content	0.79
Jiang & conrath	Information Content	0.83
Tversky]	Feature-Based	0.73
X-Similarity	Feature-Based	0.74
Rodriguez	Hybrid	0.71

8. Conclusion

Dans ce chapitre, nous avons introduit la notion de la mesure de similarité qui est une notion très importante dans le domaine de TALN et de son utilisation dans des différents domaines, On s'est basé sur la similarité statistique et sémantique et ses méthodes en détaillent. Nous avons ainsi fait un Comparaison des différentes méthodes de mesure de similarité et Les avantages et les inconvénient de chaque méthode.

Chapitre 04 :

Proposition de notre méthode

1. Introduction

Nous avons vu que la désambigüisation consiste à déterminer quel est le sens le plus approprié pour chaque mot d'un texte dans son emplacement prédéfini. Dans ce chapitre, nous allons décrire notre propre algorithme de désambigüisation basée sur les connaissances, la structure du thésaurus WordNet, combinée à l'usage d'une mesure de similarité pour déterminer un score global indiquant les connexions en termes de similarité entre les différents mots (noms) qui représentent des concepts du contexte de discours.

2. Description de notre démarche de désambigüisation

Dans le chapitre 2, nous avons donné des définitions du concept de désambigüisation, des différentes formes qui peuvent être regroupées sous la notion d'ambigüité linguistique.

L'approche que nous avons développée dans ce projet de mémoire, vise à utiliser les concepts qui constituent un contexte donné avec le mot ambigu pour décider du sens à attribuer au mot et qui devrait être le sens approprié (souhaité).

Aussi, nous avons passé en revu, différentes approches de désambigüisation existantes, avec une comparaison de leurs résultats. Notre approche se situe parmi les approches utilisant des bases de connaissances, plus précisément le thésaurus WordNet pour accomplir le processus de désambigüisation. Ce choix est motivé par le fait que WordNet [43] est une base de données lexicale de large couverture développée depuis 1985 par des linguistes du laboratoire des sciences cognitives de l'université de Princeton.

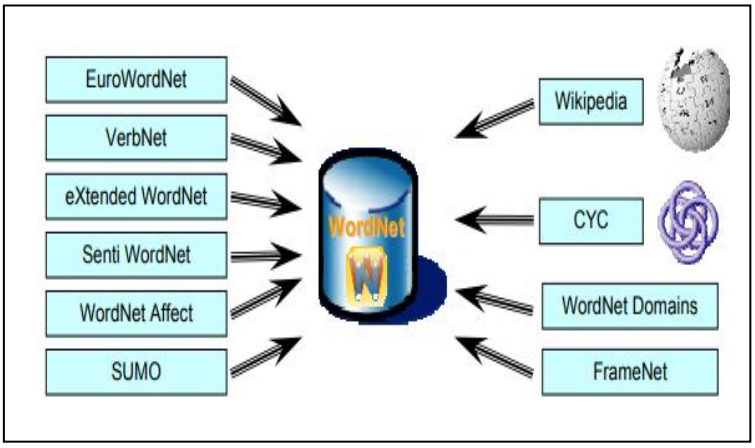

Figure 13: Ressources disposant d'une traçabilité vers WordNet**.** *[44]*

La structure de WordNet (synsets et relations hiérarchiques), comme le montre la (Figure14), offre une grande souplesse pour appliquer des mesures de similarité entre les synsets, selon différentes relations. En particulier pour dégager des associations sémantiques nécessaires au processus de désambiguïsation.

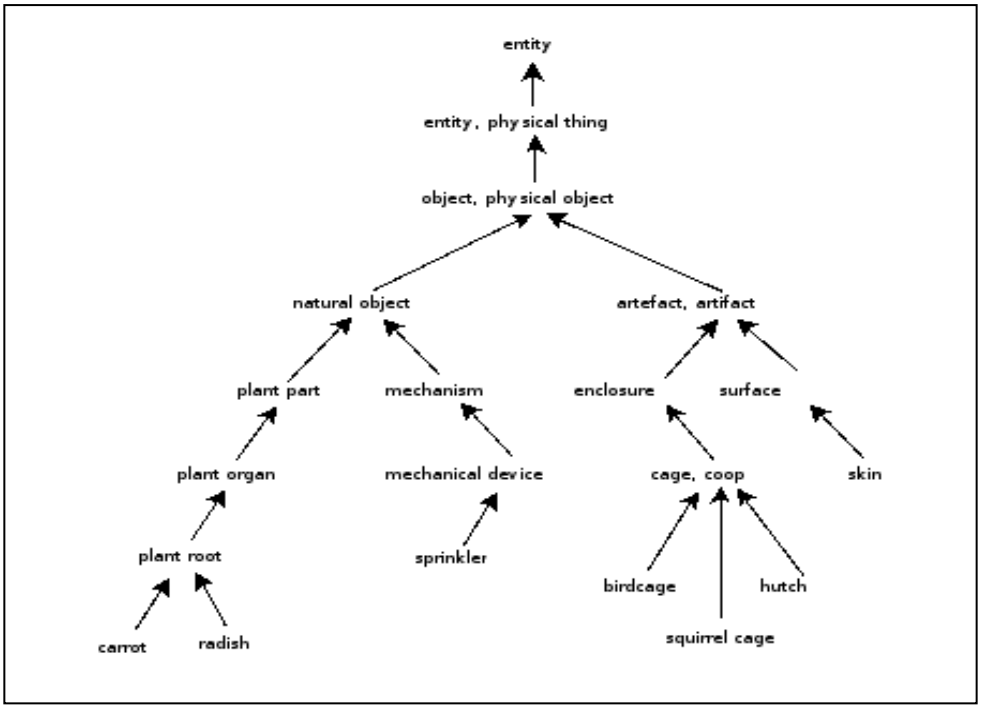

Figure 14: les relations IS-A dans WordNet

3. La Taxonomie WordNet [45]

WordNet et une taxonomie hiérarchique, une base de données lexicale développée et continue d'être mise à jour et améliorée à l'université Princeton depuis 1986. Fondée sur une théorie psychologique du langage, sa première version diffusée remonte à juin 1991.

The noun window has 8 senses (first 4 from tagged texts)

1. (72) **window** -- (a framework of wood or metal that contains a glass windowpane and is built into a wall or roof to admit light or air)
2. (6) **window** -- (a transparent opening in a vehicle that allow vision out of the sides or back; usually is capable of being opened)
3. (3) **window** -- (a transparent panel (as of an envelope) inserted in an otherwise opaque material)
4. (1) **window** -- (an opening that resembles a window in appearance or function; "he could see them through a window in the trees")
5. **window** -- (the time period that is considered best for starting or finishing something; "the expanded window will give us time to catch the thieves"; "they had a window of less than an hour when an attack would have succeeded")
6. windowpane, **window** -- (a pane of glass in a window; "the ball shattered the window")
7. **window** -- (an opening in the wall of a building (usually to admit light and air); "he stuck his head in the window")
8. **window** -- ((computer science) a rectangular part of a computer screen that contains a display different from the rest of the screen)

Figure 15: différents synsets relatifs au mot « window » dans WordNet

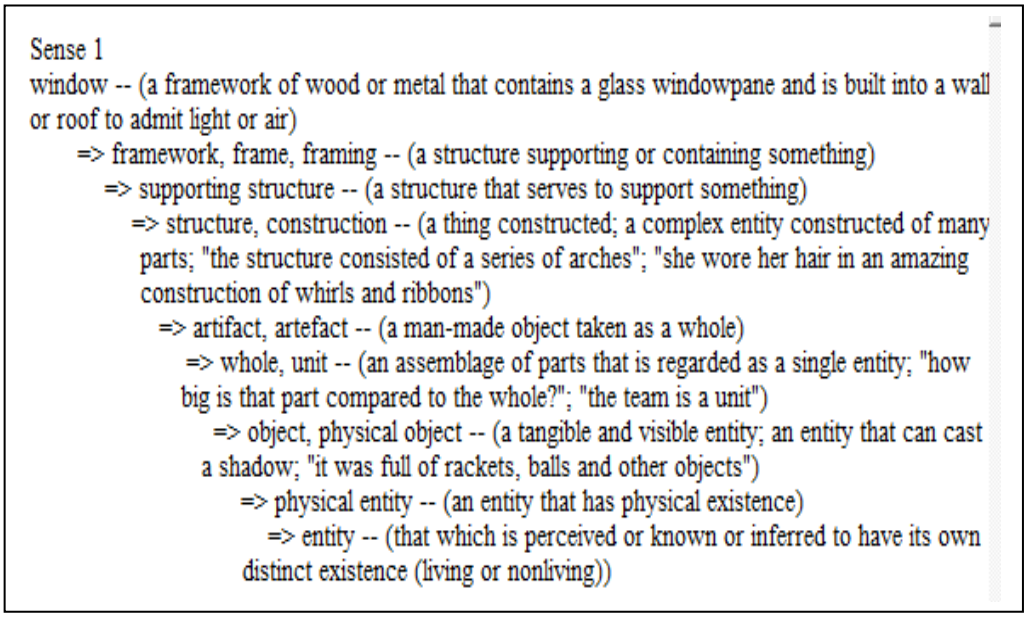

Figure 16: relation d'hyperonymie du mot « window » pour le 1er sens dans WordNet. *[45]*

Son but est de répertorier, classifier et mettre en relation de diverses manières le contenu sémantique et lexical de la langue anglaise. [46]

Des versions de WordNet pour d'autres langues existent, mais la version anglaise est la plus complète à ce jour.

Dans les figure 15 et figure 16 , nous avons un extrait de cette hiérarchie relatif aux différentes relations de synonymie , d'hyperonymie et d'hyponymie du mot « Window »

WordNet est utilisable librement par des programmes issus du monde de l'Intelligence Artificielle. L'ensemble constitue un « écosystème » complet couvrant des aspects lexicaux, syntaxiques et sémantiques. Combinées, ces ressources fournissent un point de départ intéressant pour des développements sémantiques en TAL ou dans le cadre du Web sémantique, tels que la recherche d'information, l'inférence pour la compréhension automatique de textes, la désambiguïsation lexicale.

Des interfaces de programmation sont disponibles pour de nombreux langages. Malgré certaines critiques (granularité très fine, absence de relations paradigmatiques…), WordNet est l'une des ressources de TAL la plus utilisée.

La base de données WordNet regroupe des noms, des verbes, des adjectifs et des adverbes dans des sets de synonymes, liés par des relations sémantiques (hypernymes, hyponymes, synonymes, méronymes es et holonymes qui déterminent les sens des mots. [43] [47]

On peut dons dire que cette hiérarchie offres deux services :

- Vocabulaire qui décrit les différents sens des mots
- Ontologie qui décrit les relations sémantiques entre les mots. [48] [49]

4. Choix de la mesure de similarité utilisée

Après étude et comparaison des différentes approches existantes pour mesurer la similarité sémantique entre les concepts dans WordNet, nous avons opté pour la mesure de Leacock et shodorow [50]**.** Elle basée sur la longueur du chemin et dépend donc, de la longueur du plus court chemin entre concepts comparés dans WordNet selon la relation is-a. Cette mesure est inversement proportionnelle à la profondeur de l'hiérarchie qui représente le plus long chemin de la feuille à la racine de l'hiérarchie. Cette mesure est définie comme suit :

$$\mathrm{Sim}_{\mathrm{lcha}}(c_k, c_l) = \max_I \left[-\log \left[\frac{\mathrm{length}_I(c_k, c_l)}{2D} \right] \right] \qquad (1)$$

Length_I (C_k, C_l): la longueur du chemin 'I' entre wC_k, C_l

D: la profondeur maximale de la taxonomie

C_k, C_l: Concepts à comparer

Cette mesure présente les avantages suivants :

- Simple, facile à évaluer
- Utilise la fonction logarithmique qui est une fonction convexe, avec sa courbe douce et régulière. Elle est également plus sensible aux petites variations que les fonctions linéaires, ce qui permet d'obtenir des résultats plus précis.
- Dans le chapitre 3, nous avons établi une comparaison et une évaluation des différentes méthodes de calcul de similarité, de la « Table 3 » nous avons avec cette méthode une corrélation avec le jugement humain de 82%, donc une évaluation assez appréciable.
- Selon la formule ci-dessus, plus deux concepts sont proches , plus leurs similarité augmente , et inversement plus ils sont éloignés moins ils seront similaire, c'est tout a fait adapté à la structure arborescente de WordNet.

Lorsque L (longueur du plus court chemin) → 2D alors la similarité tend vers 0, c'est-à-dire les deux concepts sont très éloignés.

Lorsque L → 2 (le plus court chemin possible entre deux concepts) nous obtenons une similarité de sim(C_k, C_l)= -log(2/2D)=Log(D), c'est la similarité maximale.

On peut donc normaliser les mesures obtenues dans [0,1] en divisant par Log(D).

5. Approche de désambiguïsation proposée

L'approche proposée repose sur des outils et des algorithmes pour déterminer le sens du mot ambigu dans le contexte donné.

1. Usage d'une base de données lexicale, dans notre cas c'est WordNet, pour l'extraction des différents sens du mot ambigu dans le contexte utilisé.
2. Usage de l'outil **NLTK, ou (Natural Language Toolkit) qui est** une suite de bibliothèques logicielles et de programmes. Elle est conçue pour le traitement naturel symbolique et statistique du langage anglais en **langage Python**. C'est l'une des bibliothèques de traitement naturel du langage les plus puissantes. Cette suite d'outils rassemble les algorithmes les plus communs du traitement naturel du langage comme le tokenizing, le part-of-speech tagging, le stemming, l'analyse de sentiment, la segmentation de topic ou la reconnaissance d'entité nommée.
3. Calcul des similarité du mot ambigu « w » avec les autres mots extrait du contexte, soient $w_1, w_2, \ldots w_n$, en utilisant la méthode de Leacock décrite en haut , donc nous obtenons $Sim(w,w_i$ i=1,n).
4. Calcul d'une série de pondération Pi et d'une série de score Si pour (i=1,n)
5. Détermination du sens du mot w, en choisissant la similarité maximale obtenue.

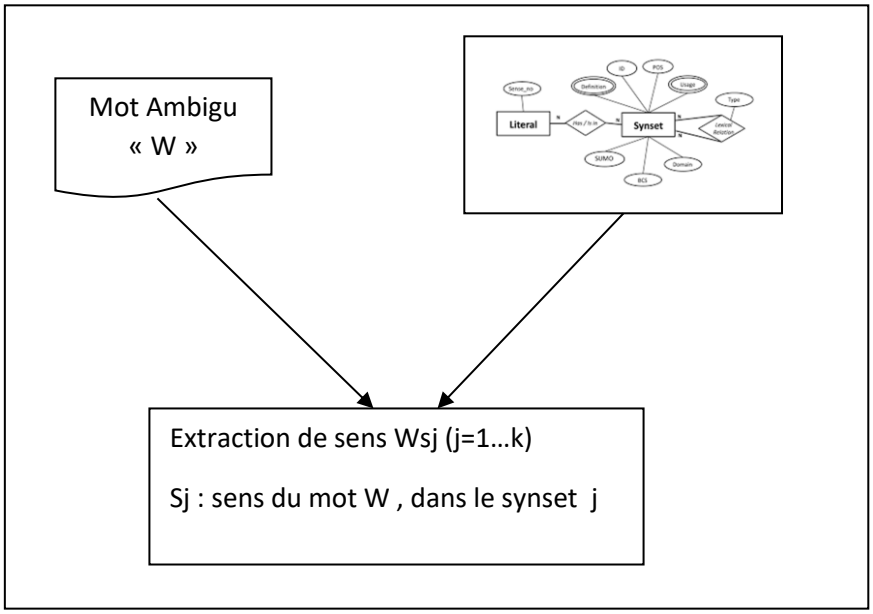

Figure 17: Extractions depuis WordNet des sens du mot ambigu "W"

Les algorithmes correspondant aux prétraitements de la phrase qui constitue donc le contexte du discours, ne fait pas partie de notre travail, donc nous supposons avoir effectué les phases de tokenizing, le part-of-speech tagging, le stemming et la reconnaissance d'entité (noms).

Le schéma ci dessus « figure 17 », décrit la première étape de notre démarche.

Etape1 : dans cette étape de la démarche, et utilisant WordNet, on exécute un algorithme qui déterminer tous les sens possibles du mot ambigu «W », on note ici que seul les synsets de « Nom » sont considérés, on ne s'intéresse pas au sens du mot en tant que verbe.

On utilisera la notation suivante, pour désigner cet ensemble de sens :

Wsj, j=1..k, pour signifier qu'il y a j sens différents liés au mot « W »

Le reste des étapes est donné pat le schéma ci-dessous, qui décrits :

- Comment calculer la similarité de chaque sens Wsj du mot W, par rapport au contexte « T » constitué de l'ensemble des mots (noms), du processus inclut :

o Détermination des profondeurs des mots du contexte dans Wordnet, et ensuite le calcul de la longueur du plus court chemin, reliant un mot « Wi » avec le sens j du mot ambigu « W », le nombre des mots dans le contexte « T » étant « i », ce processus est répété pour tous les sens « j » associé au mot ambigu « W », cette longueur est exprimée par le nombre d'arcs reliant « Wi » à « Wsj » dans l'ontologie WordNet.

o Calculer selon la formule donnée ci-dessus « formule 01 » de « Leacock et shodorow », la similarité **Sim (Wi, Wsj)** , entre les mots « Wi » à « Wsj ». cette formule utilise la longueur du plus court chemin calculé précédemment et la profondeur maximale de WordNet, ici supposée égale à 16.

$$\mathrm{Sim}(\mathrm{Wi},\mathrm{Wsj}) = -\log\left(\frac{\mathrm{Lenght(Wi,Wsj)}}{2\mathrm{D}}\right) \qquad (2)$$

o Calcul d'un poids entre tous les mots « Wi » et chaque sens « Wsj », c'est un rapport entre la profondeur du sens Ws Wj et la somme des profondeurs des autres mots Wi (i=1…n) qui forment le contexte « T » du mot ambigu « W »

Ce calcul et donné par la « formule 02 » ci-dessous

$$\mathrm{Pij(Wi,Wsj)}=\frac{\mathrm{D(Wsj)}}{\sum_{\mathrm{i=1}}^{\mathrm{n}}\mathrm{D(Wi)}} \qquad (3)$$

Pij(Wi,Wsj): poids associé au sens j du mot Wsj par rapport au mot Wi , qui est dans le contexte « T »

D(Wsj) : profondeur du mot Wsj (j=1…k) ; k sens differents du mot « W »

D(Wi) : profondeur du mot Wj (i=1…n) ; i mots du contexte « T »

o Pour chaque sens j du mot Wsj, on calcul un score Sij, ayant comme paramètres la somme des similarités calculées entre le sens du mots Wsj et les mots du contexte Wi (i=1…n)

$$\mathrm{Sij} = \mathrm{Pij} * \sum_{\mathrm{i=1}}^{\mathrm{n}} \mathrm{Sim}(\mathrm{Wi},\mathrm{Wsj}) \qquad (4)$$

La dernière étape consiste à choisir parmi tous les scores obtenus, celui ayant la valeur maximale, cela devrait exprimer l'idée que ce sens s et se basant sur la structure et les connexions des synsets dans WordNet est le plus approprié dans le contexte « T » des mots (Wi, i=1…n). Le schéma général de la démarche est donné en « Figure 18 »

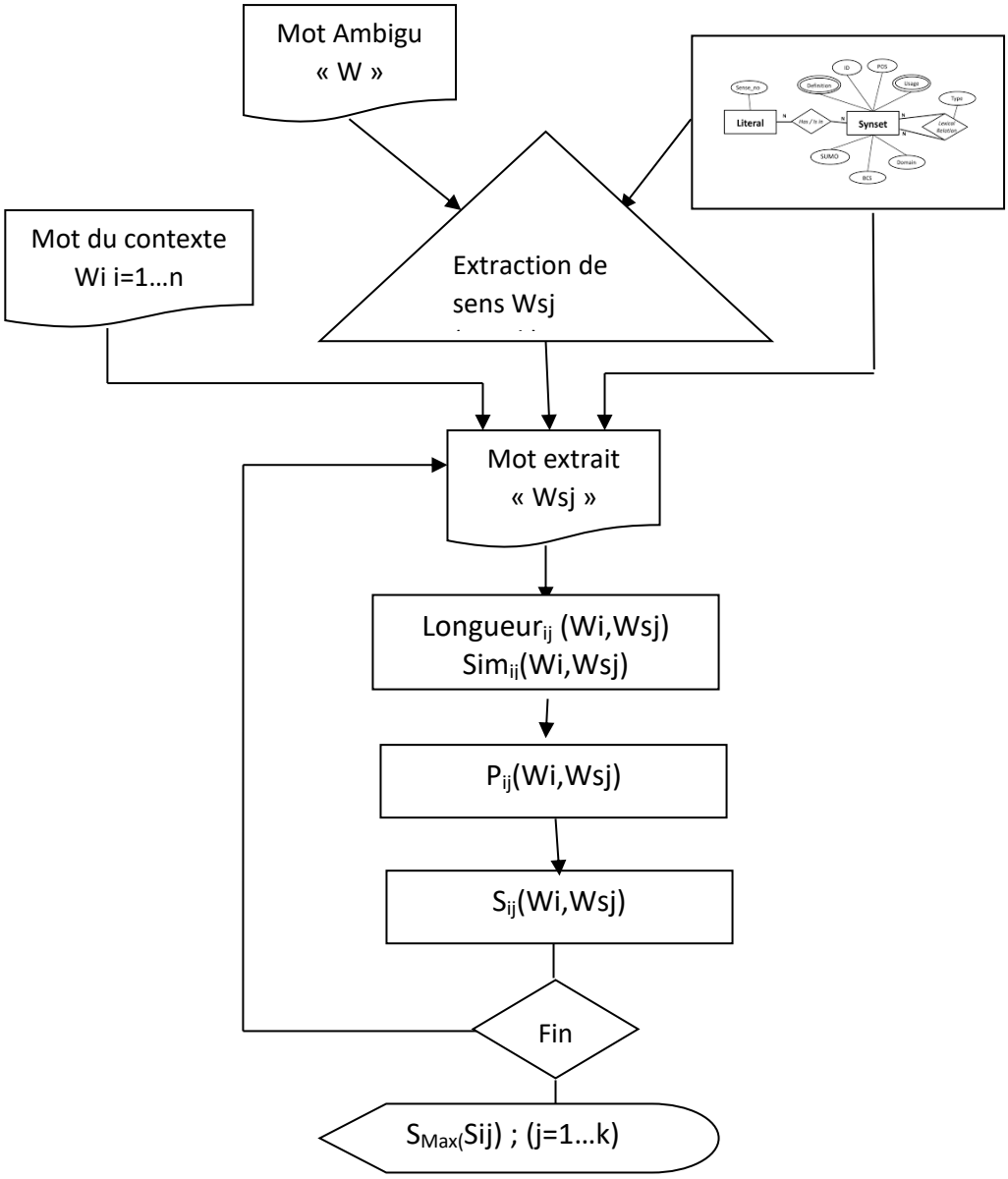

Figure 18: Algorithme général de la démarche de désambiguïsation.

6. Description de l'Algorithme

Dans cette section, nous allons donner en détail les étapes de déroulement de notre algorithme, ensuite nous verrons les outils utilisés pour expérimenter notre approche à travers l'étude de cas bien choisis, pour mettre l'accent sur la problématique abordée dans ce mémoire, qu'est la désambiguïsation.

Nous donnerons progressivement dans cette étude les explications nécessaire pour la compréhension du déroulement de cet algorithme.

Nous commencions donc par donner des détails d'exécution de l'algorithme, qui de manière global accomplit les fonctionnalités suivantes :

Début

- Lire le mot ambigu
- Récupérer ses différents sens depuis WordNet

/ sont des synsets exprimant les différents sens/

- Pour chacun des sens, calculer la profondeur du synset
- Pour tous les autres mots du contexte, calculer leurs profondeurs dans WordNet
- Pour chaque sens :
  - pour chaque mot du contexte :
    - ✓ Calculer la longueur du chemin entre chaque paire de deux mots
    - ✓ Calculer la similarité entre les deux mots, en utilisant l'expression de Leacock et Shodorow (équation 2) donnée ci-dessus.
    - ✓ Calculer le facteur de pondération, comme donné par l'équation 3
    - ✓ Calculer le Score, selon l'équation 4
  - Déterminer le score Maximal qui correspondra au sens le plus proche et le plus pour probable d'être le sens désiré pour le mot ambigu dans le contexte actuel.
- Fixer le sens attribué

Fin

7. Les outils d'implémentation

7.1. Le langage Python

La première version du langage de programmation Python est apparu en 1991, c'est un langage interprété, c'est-à-dire qu'il ne nécessite pas une phase de compilation, il est plutôt simple, efficace et souple pour écrire des programmes traitant divers sujets.

En plus de l'écriture de simples scripts, et disposant de plusieurs bibliothèques, il permet de monter des programmes plus complexes.

* pour le Web: **python** combiné avec le **framework** Django est une bonne solution pour le développement de sites web.

* Système: Python est également utilisé pour l'administration des systèmes, avec aussi des combinaisons avec JAVA.

* **bioinformatique**. Des librairies sont disponibles pour ce domaine comme le module **biopython** .

* La création de jeux vidéo en 2D (et 3D) exemple: **pyGame** .

* Les recherches scientifiques dans presque tous les domaines d'actualité

7.2. Python Vs autres langages

Python favorise la programmation impérative structurée, fonctionnelle et orientée objet. Il est doté d'un typage dynamique fort, il présente l'avantage d'être facile à apprendre, avec un code très lisible et facilement modifiable, ce qui évite les bugs. En plus il dispose de plusieurs librairies très riches. Aussi la doc**umentation python** est disponible aidant à dépasser certaine difficultés.

Dans les applications de l'intelligence artificielle Python est très utilisé, comme pour implémenter et expérimenter des algorithmes d'apprentissage automatique et de deep learning, en particulier grâce aux frameworks tels que **TensorFlow** et **PyTorch**.

Parmi les bibliothèques utilisées en traitement des langages naturels, nous avons NLTK

La bibliothèque NLTK (Natural Language Toolkit**)**.

C'est est une bibliothèque Python dédiée au traitement naturel du langage (NLP). Notre problématique est évidemment située au cœur des algorithmes traitant les langages naturels, l'ambiguïté des langages naturels. (Figure 19)

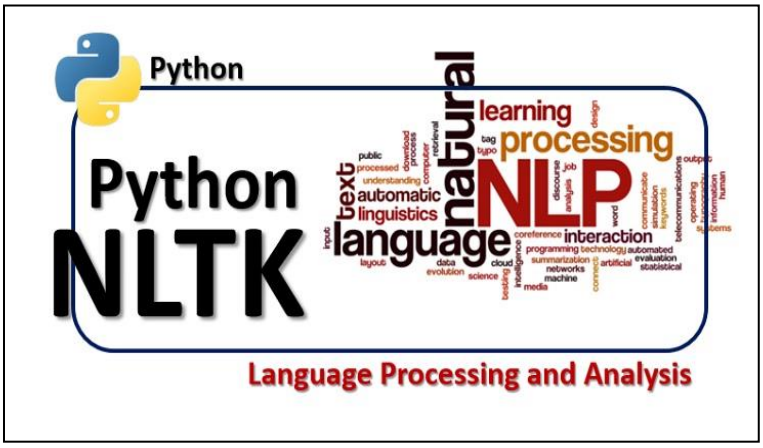

Figure 19: usages de NLTK dans Python

L'ambiguïté se manifeste par la multitude d'interprétations possibles pour un niveau de traitement, comme : – l'ambiguïté de la Polysémie, que nous traitons dans ce travail.

NLTK est une suite de bibliothèques logicielles et de programmes. Elle est conçue pour le traitement naturel symbolique et statistique du langage anglais en langage Python. C'est l'une des bibliothèques de traitement naturel du langage les plus puissantes.

NLTK, inclut **les algorithmes les plus utilisés dans le NLP,** comme le tokenizing, le **part-of-speech tagging**, le **stemming, l'analyse de sentiment**, la **segmentation de topic** ou la **reconnaissance d'entité nommée**.

7.3. Configuration matérielle

Modèle : HP PC G3 Notebook

Type : PC à base de x64

Processeur : Intel(R) Core (TM) i3-4005U CPU @ 1.70GHz, 1701 MHz, 2 cœur(s), 4

Mémoire physique (RAM) installée 4,00 Go

Mémoire physique totale 3,94 Go

Mémoire virtuelle totale 7,31 Go

Un disque dur fixe

Partitions : 4

Capacité : 465,76 Go

7.4. Configuration Logicielle

- Système d'exploitation : Microsoft Windows 10 Professionnel

Version : 10.0.19045 Build 19045

- Python : version 3.11 et différents scripts.
- Visual studio code
- NLTK package

8. Etudes de cas et résultats obtenus

- Les cas étudiés ci-dessous ont été judicieusement choisis, pour montrer l'efficacité de la démarche, à savoir assigner le bon sens au mot ambigu dans un texte, ou une phrase donnée.
- Les mesures utilisées sont prises depuis WordNet, et sont :
  - Les profondeurs des mots
  - Les longueurs des chemins
  - Les sens d'un mot ambigu

Ces calculs ont été faits par des scripts indépendants, pour faciliter l'obtention de résultats et éviter à ce stade la complexité de la programmation dans des environnements, nécessitant un apprentissage long pour maitriser tous les outils et les méthodes disponibles.

Nous avons choisi pour expérimenter la méthode quelques phrases contenants des mots ambigus, pour voir les sens qui vont être sélectionnées et leurs pertinences par rapport
Au sens de la phrase entière.

Le tableau 4, suivant montre les phrases soumises, les sens sélectionnés et le pertinence par rapport aux résultats donnés par Google (Google traduction pour lever l'ambiguïté).

Table 4: les phrases soumises, les sens sélectionnés et le pertinence par rapport aux résultats donnés par Google.:

N°	Phrase	Mot ambigu	Les sens relevés WordNet(Sunsets)	Le sens Sélectionné	Pertinence
01	The Curator saw a **TEAR** in the Painting	**Tear**	**Tear.n.01** **Rip.n.02** **Bust.n.04** **Tear.n.04**	**Rip.n.02** (Déchirure)	100%
02	Our Housekeeper Caught a **MOUSE** in the Garden of the House	**Mouse**	**Mouse.n.01** **Shiner.n.01** **Mouse.n.03** **Mouse.n.04**	**Mouse.n.01** (Souris)	100%
03	yesterday I reserved a **TABLE** at my favorite restaurant	**Table**	**Table.n.01** **Table.n.02** **Table.n.03** **Mesa.n.01** **Table.n.05** **Board.n.04**	**Table.n.01** Table étude **Table.n.02** Manger	50% 50%

Phrases sélectionnées pour l'étude de cas.

Etude de cas 01

Nous allons exécuter les scripts qui modélisent notre démarche, sur un exemple d'une phrase contenant un mot ambigu et deux autres mots constituant le contexte.

Phrase : The Curator saw a Tear in the Painting

Mot ambigu (Polysémie) : **Tear**

Mots du contexte (Noms) : **Curator** et **Painting**

Dans la figure suivante, nous avons un aperçu de l'exécution du script « PrgWordSenses » , qui retourne le nombre et les différents noms et définitions des sens extrait de WordNet relatifs au mot ambigu qui est saisi en entrée.

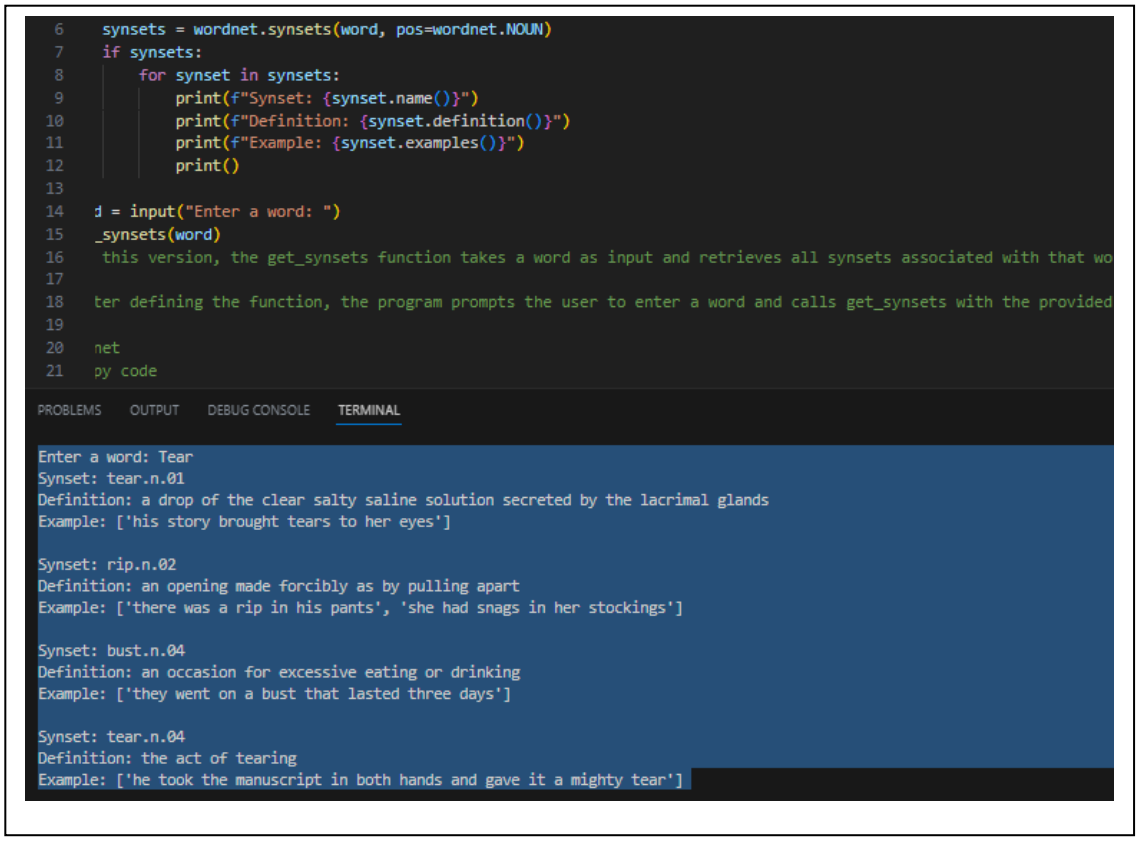

Figure 20: Extraction des sens de mot ambigu "Tear"

Pour résumer ces résultats nous avons dans le tableau suivant, les sens et les connexions entre les synsets, des mots constituant le contexte du discours

Table 5: sens du mot ambigu et connexions des synsets

TEAR(Larme)	Profondeur	RIP(déchirure)	Profondeur
Tear.n.01	6	Rake.n.01	6
Rip.n.02	6	Rip.n.02	6
Bust.n.04	10	Rip.n.03	7
Tear.n.04	9	Rent.n.04	10
BUST	**Profondeur**	**PAINTING**	**Profondeur**
Flop.n.03	6	Painting.n.01	8
Female chest.n.01	6	Painting.n.02	8
Bust.n.01	7	Painting.n.03	10
Bust.n.04	10	Painting.n.04	8
CURATOR	**Profondeur**		
Curator.n.01	7		

Dans la suite nous avons à exécuter les scripts suivants :

- LengthPathW1W2 : détermine le plus court chemin entre les mots W1 et W2, en termes de nombre d'arcs reliant les mots
- Calcul de similarités, et scores

Dans les tableaux ci-après, nous avons les résultats de cette simulation

Profondeurs : extraites depuis la taxonomie WordNet

Table 6: profondeur des mots du contexte

	Tear	Painting	Curator
Sens 1 =Tear	6	8	7
Sens 2= Rip	6	8	7
Sens 3=Bust	10	8	7
Sens 4= Tearing	9	8	7

Le tableau 7, donne la longueur des chemins entre les mots, calculés dans la hiérarchie WordNet

Table 7: longueurs des chemins entre les synsets

	Longueur chemin (Tear, Painting)	Longueur chemin (Tear, Curator)
Sens 1 =Tear	14	13
Sens 2= Rip	10	11
Sens 3=Bust	18	17
Sens 4= Tearing	17	16

Table 8: Calculs de similarités et des scores de sens

	Sim (Tear, Painting)	Sim (Tear, Curator)	Facteur de Pondération	Score
Sens 1 =Tear	0,3590	0,3912	0,4000	0,3001

Sens 2= Rip	0,5051	0,4638	0,4000	**0,3876**
Sens 3=Bust	0,2499	0,2747	0,6667	0,3497
Sens 4= Tearing	0,2747	0,3010	0,6000	0,3454

Le score maximal étant celui mentionné en rouge, il détermine de ce fait le sens le plus approprié pour le mot ambigu « TEAR » dans le contexte de la phrase donnée.

Ce sens correspond donc à « RIP » qui veut dire « Déchirure » , et donc c'est comme ci la phrase est équivalente à dire :

Phrase résultat: The Curator saw a Rip in the Painting

Ce résultat est satisfaisant, accepté du fait que la soumission de la phrase initiale à une traduction par le traducteur Google , produit le même résultat comme le montre la figure 21.

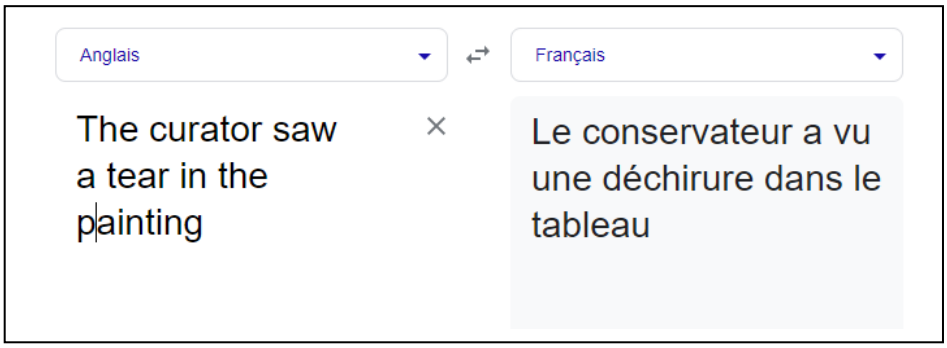

Figure 21: Traduction de Google du mot TEAR dans le contexte donné

Etude de cas 02

De même, nous ferons avec la phrase :

Phrase : Our Housekeeper Caught a Mouse in the Garden of the House

Mot ambigu (Polysémie) : **Mouse**

Mots du contexte (Noms) : **Housekeeper, Garden** et **House**

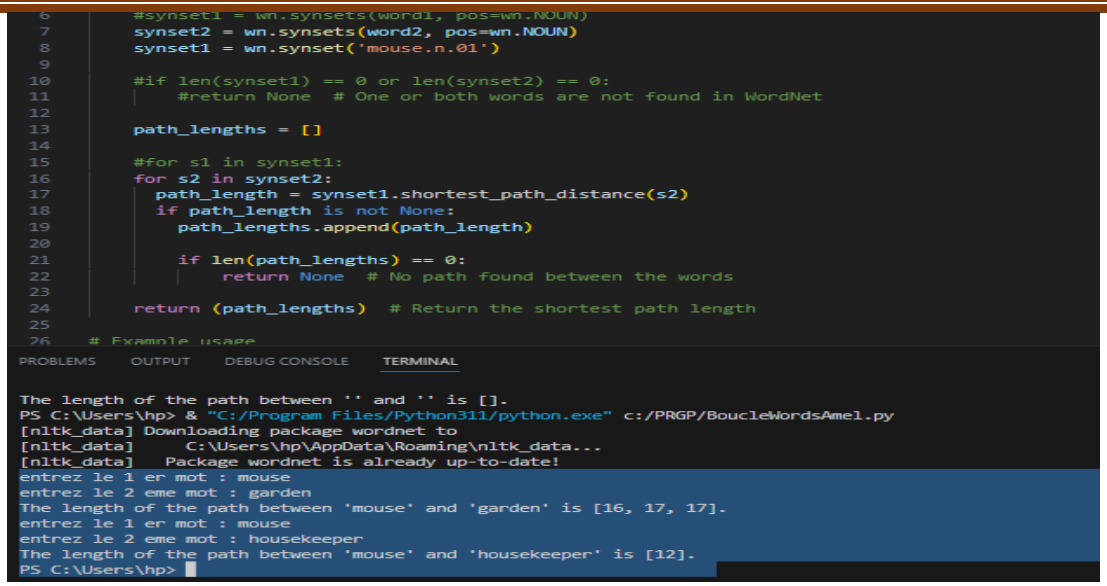

Figure 22: Script de calcul de la longueur de chemin entre deux synsets de WordNet.

Dans la Figure 22, ci-dessus, nous avons un aperçu de l'exécution du script « BoucleWordsAmel », qui prend en entrée deux mots représentant deux synsets de WordNet

Et retourne la longueur entre ces deux synsets, exprimée par le nombre d'arcs qui les relient.

Dans le tableau suivant, nous avons les sens et les connexions entre les synsets, des mots constituant le contexte du discours

Table 9: sens du mot ambigu et connexions des synsets

MOUSE	Profondeur	SHINER	Profondeur
mouse.n.01	12	shiner.n.01	10
shiner.n.01	10	shiner.n.02	3
mouse.n.03	4	common_mackerel.n.	16
mouse.n.04	8	01	16
		shiner.n.04	
HOUSEKEEPER		**GARDEN**	
housekeeper.n.01	7	garden.n.01	8
		garden.n.02	5
		garden.n.03	9
HOUSE			
house.n.01	7		
firm.n.01	7		
house.n.03___house.n.	6,6,7,8,8		

07 **sign_of_the_zodiac.n.0** **1** **house.n.09** **family.n.01** **theater.n.01** **house.n.12**	**5** **7** **6** **7** **7**		

Dans les tableaux ci-après, nous avons les résultats de cette simulation

Profondeurs : extraites depuis la taxonomie WordNet

Table 10: profondeur des mots du contexte

	Mouse	Housekeeper	Garden	House
Sens 1 = Small rodent	12	7	9	7
Sens 2= Shiner	10	7	9	7
Sens 3= Person who is quiet/ timid	4	7	9	7
Sens 4= computer mouse	8	7	9	7

Longueur des chemins entre les mots, calculés dans la hiérarchie WordNet

Table 11: longueurs des chemins entre les synsets

	Longueur (Mouse,Housekeeper)	Longueur (Mouse, Garden)	Longueur (Mouse, House)
Sens 1 = Small rodent	12	17	13
Sens 2= Shiner	17	19	17
Sens 3= Person quiet/ timid	5	11	8
Sens 4= computer mouse	12	13	7

Table 12: Calcul de similarités et scores

	Sim (Mouse,Housekeeper)	Sim (Mouse,Garden)	Sim (Mouse,House)	Facteur de Pondération	Score
Sens 1 = Small	0,4260	0,2747	0,3912	0,5217	**0,5697**

rodent					
Sens 2= Shiner	0,2747	0,2264	0,2747	0,4348	0,3373
Sens 3= Person quiet/	0,8062	0,4638	0,6021	0,1739	0,3256
Sens 4= computer M	0,4260	0,3912	0,6601	0,3478	0,5138

Le score maximal étant celui mentionné en rouge, il détermine de ce fait le sens le plus approprié pour le mot ambigu « MOUSE » dans le contexte de la phrase donnée.

Ce sens correspond donc à « Small Rodent » qui veut dire « Petit Rongeur » , et donc c'est comme ci la phrase est équivalente à dire :

Phrase résultat: Our Housekeeper caught a Small Rodent in the Garden of the House

Ce résultat est satisfaisant, accepté du fait que la soumission de la phrase initiale à une traduction par le traducteur Google, produit le même résultat comme le montre la figure 23.

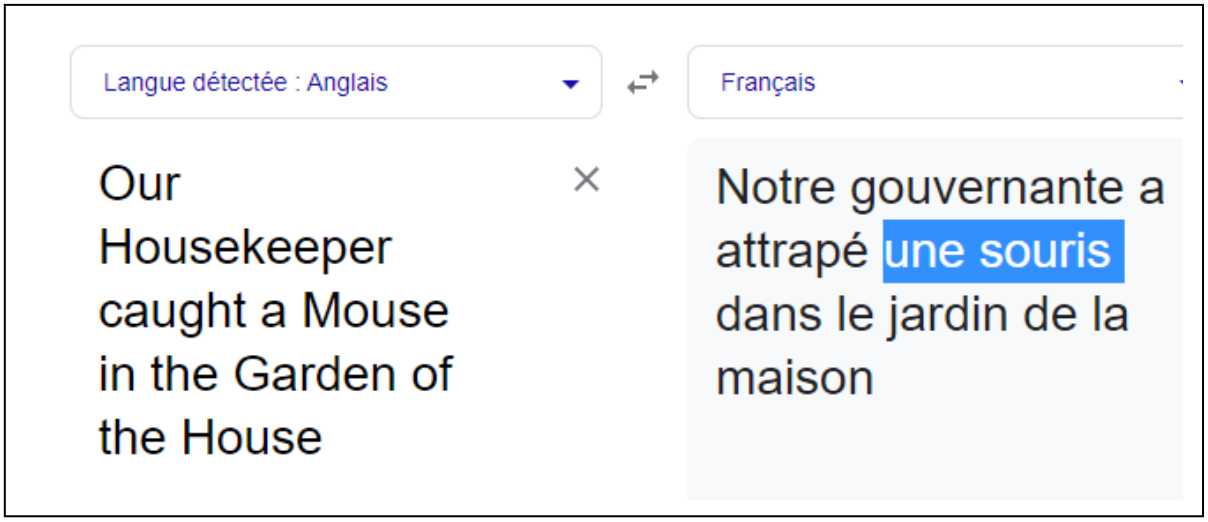

Figure 23: Traduction de Google du mot MOUSE dans le contexte donné.

Voici quelques-unes des capture d'écran de notre application :

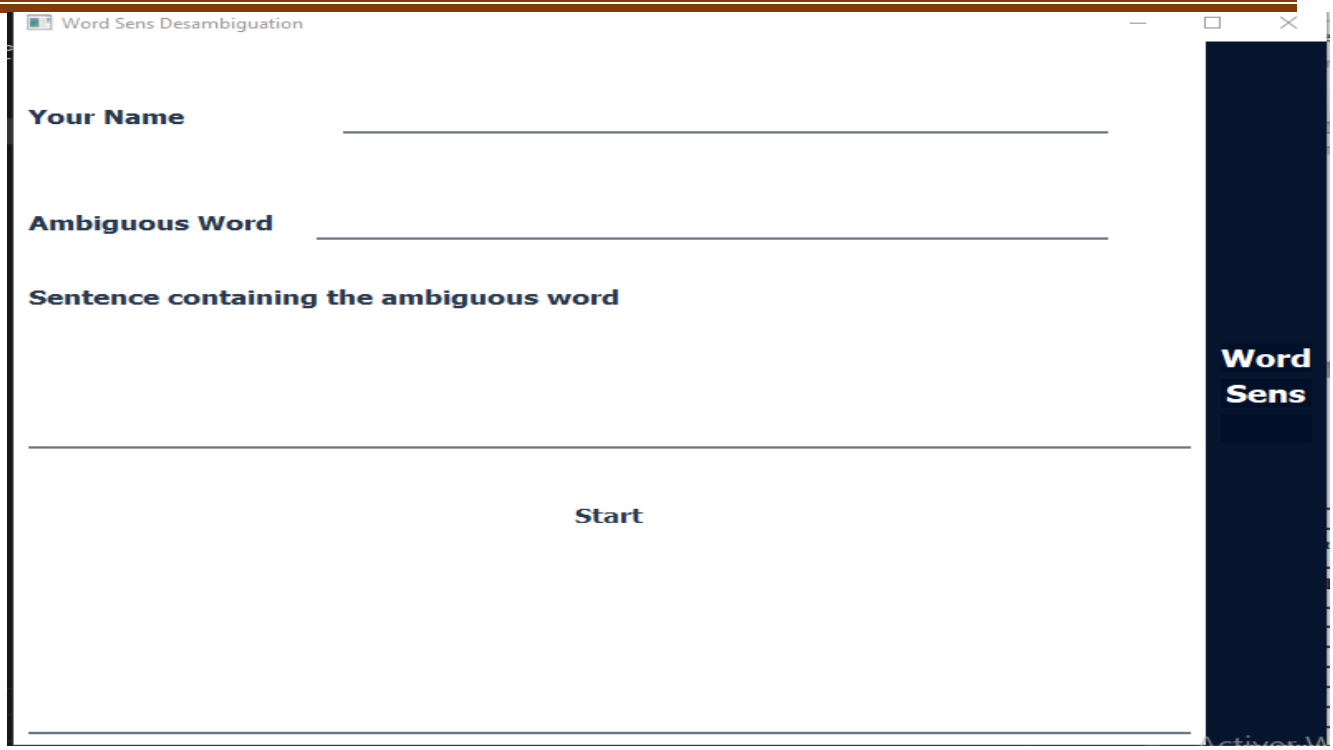

Figure 24: interface de notre application

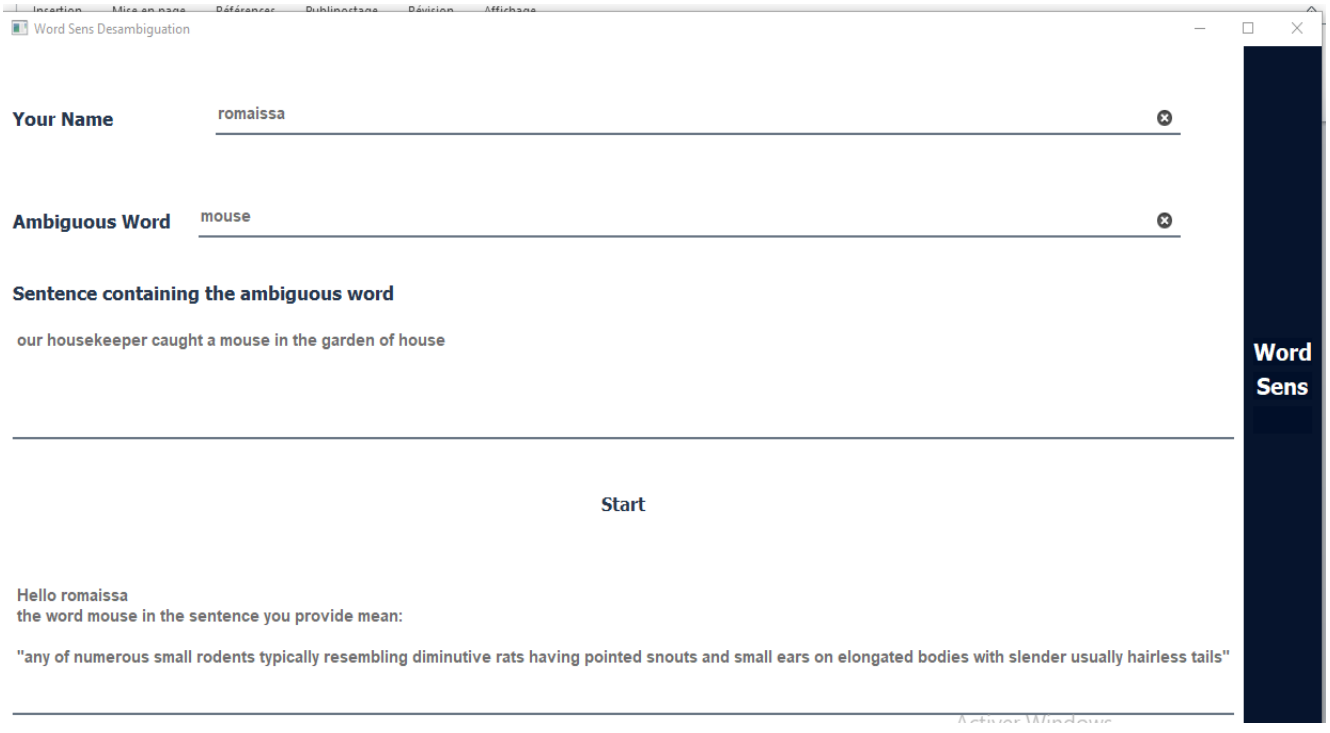

Figure 25: résultat de l'application.

9. Conclusion

Dans ce chapitre nous avons décrit les aspects fonctionnels et expérimentaux de l'approche proposée. Au début nous avons donné des définitions en relation avec la base de connaissances sur laquelle repose notre approche, qu'est WordNet. Des détails nécessaires à la compréhension de la démarche ont aussi pris part dans ce chapitre comme les synsets et leurs interconnexions.

Aussi, pour l'implémentation de cet algorithme, on a listé tous les outils utilisés comme le langage PYTHON, le package NLTK, l'outil VISUAL STUDIO et tous script nécessaire pour la création et l'utilisation de cet environnement.

Nous n'avons pas entièrement terminé le développement des programmes en une seul package installable, mais on a plutôt concentré nos efforts sur l'obtention des résultats justifiant la démarche. Au total nous avons expérimenté six (6) études cas qui ont donné 5/6 de cas avec des résultats acceptables en comparaison avec le traducteur GOOGLE, c'est-à-dire avec un taux d'acceptation de 83%.

Le cas qui ne répondait pas aux résultats attendus, relève plutôt d'une autre discipline linguistique, "Le sens figuré des mots " qui n'est pas abordé dans ce travail.

Conclusion general et Perspectives futures

La désambiguïsation de sens des mots, au sens général du mot, est une tâche assez complexe à accomplir, elle est rattachée aux recherches du domaine de traitement automatique des langages TALN.

Le but est de prédire le sens des mots dans un contexte donné, en se basant sur des ressources lexicales (des bases de connaissances) ou des Data set pour les méthodes utilisant les différentes techniques du "Learning"

Dans notre mémoire, nous avons choisi d'utiliser la ressource WordNet, un dictionnaire électronique très riche, notamment pour la langue Anglaise, de libre utilisation et offre plusieurs services aux utilisateurs des applications du TALN.

En effet, WordNet par sa structure hiérarchique, aide à déterminer le sens d'un mot dans un contexte, à partir d'un inventaire de sens prédéfini.

Dans notre démarche, nous nous somme concentré sur la désambiguïsions des termes qui représentent des noms, les adjectifs, verbes et adverbes ne sont pas considérés. Ce choix est motivé par le fait que le lexème de noms sont les plus porteurs de sens, et de ce fait sont les plus utiles au processus de désambiguïsation. Aussi, nous avons pris en considération les paramètres de calculs de similarité entre les différents noms présents dans le contexte donné. Ainsi les longueurs des chemins entre les noms, le choix de la méthode de calcul de similarité (Leacock et Shodorow), leurs profondeurs et leurs interconnexions ont été des facteurs déterminants pour la précision des scores calculés.

Les expérimentations sont réalisées en utilisant des packages récents, de large utilisation et le langage de programmation "Python " efficace pour ce type d'application de l'intelligence artificielle. Les résultats obtenus ont été très encourageant, montrant une efficacité acceptable et prometteuse. Ainsi, des améliorations seront susceptibles d'apporter de meilleurs résultats. En fait, la structure de la taxonomie utilisée, WordNet, a été bien exploitée pour localiser les positions des mots dans cette hiérarchie, pour connaitre les relations d'interconnexion, calculer les distances entre les mots pour enfin déterminer un score avec la formule que nous avons proposée.

Les perspectives futures de ce travail, vont dans le sens d'élargir le syntagme nominal, au syntagme verbal et adjectival. Certes, cette généralisation produirait plus de précision dans les résultats, voire même la possibilité de traiter des aspects plus compliques comme le sens figuré des mots grâce aux fonctions des verbes et des adjectifs.

Bibliographies

Bibliographie

[1] Y. MEZAOUR et A. HAFID, *(Désambiguïsations des termes ambigus dans des documents textuels), Memoire de master Informatique,UNIVERSITE MOULOUD MAMMERI DE TIZI-OUZOU,,* 2016.

[2] Z. BENBLAL et F. BELOUAFI , *(Intégration d'un lemmatiseur arabe dans le cadre), Thème de mémoire de master, Université Ahmed Draia - Adrar,* 2014/2015.

[3] T. Y. BOUGHERARA , *BOUGL'utilisation du traitement automatique des langues (T.A.L.) pour l'étude des adverbes français dans les textes journalistiques : modalisation et subjectivité), Mémoire de magister, Université d‟Oran 2,,* 2015/2016.

[4] F.YVO, *Une petite introduction au traitement Automatique du langage naturel,* 2007.

[5] A. Laurent, *(Traitement Automatique du Langage Naturel (TALN) Outils d'analyse de données textuelles), Université Paris 13 – Laboratoire d'Informatique de Paris-Nord (LIPN),* 4 novembre 2010.

[6] M. FABIEN, «(Traitement Automatique du Langage Naturel en français (TAL / NLP)),» 3 11 2019. [En ligne]. [Accès le 24 12 2022].

[7] P. Langlais, *(INTRODUCTION AU TALN), universété de Montréal,* 12janvier 2014.

[8] M. Dr.Loukam, *chapitre 01 introductin TALN, support de cours.*

[9] B.-E. BENAISSA, *(Construction semi-automatique d'ontologies à partir de textes arabes), Mémoire de magister, Université Abou Bakr Belkaid,* BENAISSA Bedr-Eddine, « Construction semi-automatique d'ontologies à partir de textes arabes », Mémoire2011/2012.

[10] J. CHALON, *(LES APPLICATIONS DU TRAITEMENT AUTOMATIQUE DU LANGAGE NATUREL), Diplome Superieur de Bibliothecaire, Universite Claude Bernard Lyon 1,* 1991.

[11] G. BENALIOUA, *(Le Deep Learning pour la Reconnaissance Automatique de la Parole) thème de mémoire de master, Université Saida Dr. MOULAY Tahar,* Juin 2019.

[12] R. BOUHACI et T. SLIMANI, *(Indexation temporelle d'une ontologie lexicale),mémoire de master en informatique,Université Mouloud Mammeri -Tizi Ouzou-,* 2016/2017.

[13] H. BENSMAIN , *(Traitement des données dans le web sémantique par des Graphes), Rapport du mini projet, Université Abdelhamid Ibn Badis,* 2015/2016.

[14] B. FOGUEM, V. CHAPURLAT et F. PRUNET, *(MODELISATION ET ANALYSE FORMELLE DES SYSTEMES COMPLEXES AVEC LES GRAPHES CONCEPTUELS : APPLICATION À LA MODELISATION D'ENTREPRISE)a.*

[15] D. DJEBBAR, *(la representation graphique des connaissances : les réseaux sémantiques), Support de cour, Université Annaba,* 2017/2018.

[16] [En ligne]. Available: https://www.definitions360.com/ambiguite/. [Accès le 14 02 2023].

[17] C. Fuch, 1996. [En ligne]. Available: https://www.erudit.org/fr/revues/rql/1999-v27-n1-rql2948/603171ar.pdf. [Accès le 07 02 2023].

[18] A. BERROUBI et S. BEN LATRACHE , *(Vers un Système pour le Résumé Automatique), Mémoire de master en informatique, Université Mohamed BOUDIAF - M'SILA-,* 2021/2022.

[19] T. Andon , «État de l'art : mesures de similarité sémantique locales et,» pp. 295-308, 2012.

[20] C. Fuch, «L'ambiguïté: du fait de langue aux stratégies,» pp. 5-18, 2009.

[21] S. DJABALLAH , *(Essai sur l'ambiguïté syntaxique en linguistique générale; étude analytique), Mémoire de master en langue français, Université Kasdi Merbah -Ouargla-,* 2013/2014.

[22] «lemagit.fr,» [En ligne]. Available: https://www.lemagit.fr/definition/Ambiguite-lexicale. [Accès le 15 02 2023].

[23] F. Bédard, H. Bodson et J. Hould-Fortin, «LE TRAITEMENT DES AMBIGUÏTÉS SYNTAXIQUES EN CONTEXTE,» pp. 80-107, 2011.

[24] «lemagit.fr,» [En ligne]. Available: https://www.lemagit.fr/definition/Ambiguite-structurelle. [Accès le 15 02 2023].

[25] «definition 360.com,» [En ligne]. Available: https://www.definitions360.com/ambiguite/. [Accès le 14 02 2023].

[26] L. Vial, B. Lecouteux et D. Schwab, «Représentation vectorielle de sens pour la désambiguïsation,» p. 142–149, 6 2017.

[27] B. Mokhtar Boumedyen et G. Núria , «Approches d'analyse distributionnelle pour améliorer la désambiguïsation sémantique,» 06 2016.

[28] M. Laroussi , Z. Anis et Z. Mounir, «Approche basée sur les arbres sémantiques pour la désambiguïsation lexicale de,» pp. 281-290, 2014.

[29] M. B. Billami, «Désambiguïsation lexicale à base de connaissances par sélection distributionnelle,» pp. 13-24, Juin 2015.

[30] K. Bousmaha, S. Charef_Abdoun, L. Hadrich_Belguith et M. Rahmouni, «Une approche de désambiguïsation morpho_lexicale évaluée sur l'analyseur morphologique Alkhalil,» pp. 26-40, 05 03 2013.

[31] N. Sharma et P. S. Niranjan, «Applications of Word Sense Disambiguation: A Historical Perspective,» 2015.

[32] «engati.com,» [En ligne]. Available: https://www.engati.com/glossary/word-sense-disambiguation. [Accès le 04 04 2023].

[33] «devopedia.org,» 28 Juin 2021. [En ligne]. Available: https://devopedia.org/word-sense-disambiguation. [Accès le 04 04 2023].

[34] I. BENACHOUR et H. CHIKHAOUI , *(Catégorisation automatique des textes avec des mesures de similarité sémantique), Mémoire de master en informatique, Université Abou Bakr Belkaid-Tlemcen-,* 2018/2019.

[35] M. RIFQI, *(Mesures de similarité, raisonnement et modélisation de l'utilisateur), HABILITATION A DIRIGER DES RECHERCHES DE L'UNIVERSITE PIERRE ET MARIE CURIE,* 2010.

[36] S. TOBBI, A. CHERIET et D. Nessah, *(Modeling and experimentation of a new measure of semantic similarity in a conceptual hierarchy), Mémoire de master en informatique,Université Abbes Laghrour -khenchela-,* 2021/2022.

[37] A. SIAGH et C. DEROUICHE, *(Similarité sémantique entre concepts : Application à la recherche d'images),*

Université Kasdi Merbah - Ouargla-.

[38] A. Tchechmedjiev, «État de l'Art : Mesures de Similarité Sémantique Locales et Algorithmes Globaux pour la Désambiguïsation Lexicale à Base de Connaissances,» pp. 295-308, Juin 2012.

[39] D. Nessah, O. Kazar et A.-N. Benharkat, «Towards a hybrid semantic similarity measure to set the conceptual relatedness in a hierarchy,» pp. 155-164, 2016.

[40] A. N. Ngom, «Etude des mesures de similarité sémantique basées sur les arcs,» pp. 535-544, 2015.

[41] G. Miller et . W. Charles , «Contextual Correlates of Semantic Similarity. Language and Cognitive Processes,» pp. 1-28, 1991.

[42] E. Petrakis, G. Varelas, A. Hliaoutakis et P. Raftopoulou, «X-Similarity: Computing Semantic Similarity between Concepts from Different Ontologies,» *Digital Information Management,* pp. 233-237, 2006.

[43] G. A. e. al, *(WordNet : A lexical database for English), ACM, V38/11,* 1995.

[44] . C. François-Régis , *Fran (WordNet et son écosystème : un ensemble de ressources linguistiques de large couverture) BDL-CA, Montréal,,* avril 2007.

[45] «WordNet,» [En ligne]. Available: https://wordnet.princeton.edu. [Accès le 02 06 2023].

[46] Y. MEZAOUR , H. Arezki et S. Iltache, *] Younes MEZAOUR Ar(Désambiguïsations des termes ambigus dans des documents textuels), Mémoire Master, Université Tizi ouzou,* 2016.

[47] A. Domagoj, S. Jian et S. Pado, «(Leveraging Lexical Substitutes for Unsupervised Word Sense Induction.) AAAI,,» pp. 5004-5011, 2018.

[48] D. Nessah et O. Kazar , «(An improved semantic information searching scheme based multiagent system and an innovative semantic similarity measure),» pp. 288-297, 2013.

[49] . B. Alexander et . H. Graeme, «Alexander(Evaluating WordNet-based Measures of Lexical Semantic Relatedness.), Comput. Linguistics 32(1),,» pp. 13-47, 2006.

[50] C. Leacock et M. Chodorow, «Leacock C.(Combining Local Context and WordNet Similarity for Word Sense Identification in WordNet." In: Christiane Fellbaum, editor, An Electronic Lexical Database,,» pp. 265-283, 1998.

[51]

[52] O. Pr.KAZAR, *(Représentation des connaissances et raisonnements) , Support de cour Université de Biskra.*