الجمهـــورية الجزائـــريـــة الديمـــقـــراطيـــة الـــشعـــبـــيـــة

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليـــم العالي و البحث العـــلمـــي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جـــامعة أبي بكـــر بلقـــايد – تـــلمسان –

Université Aboubakr Belkaïd – Tlemcen –

Faculté des SCIENCES

MEMOIRE

Présenté pour l'obtention du **diplôme** de **MASTER**

En : INFORMATIQUE

Spécialité : Système d'Information et de Connaissance

Par : BEKARA INES

Sujet

Réalisation d'un système de recherche d'information à base d'appariement sémantique

Soutenu publiquement, le 26 / 06 / 2023 , devant le jury composé de :

Mme KHITRI Souad	Président
M BENAISSA Mohamed	Examinateur
M BENTAALLAH Mohamed Amine	Encadrant

Remerciements

Au terme de ce travail, je tiens tout d'abord à remercier Dieu le tout puissant et miséricordieux qui m'a donné la force et la patience durant ces longues années d'étude.

Je tiens à exprimer ma profonde gratitude et mes sincères remerciements à mon encadrant Dr. ***Bentaallah Mohamed Amine*** *pour son soutien, sa patience ses précieux conseils, son aide, sa disponibilité tout au long de mes études et sans qui ce mémoire n'aurait jamais vu le jour.*
Qu'il trouve dans ce travail un hommage vivant à son grand dévouement et à sa haute personnalité.

Je tiens tout particulièrement à remercier les enseignants du département d'informatique pour leur disponibilité et encouragement, ainsi que tous les enseignants qui ont contribué à notre formation.

Ma reconnaissance va aussi aux membres de jury, pour l'honneur qu'ils auront fait en acceptant de juger ce travail.

Je remercie, enfin tous ceux qui, d'une manière ou d'une autre, ont contribué à la réussite de ce travail et qui n'ont pas pu être cités ici

Dédicace

Cette dédicace est un hommage à vous tous, les piliers qui ont façonné ma vie et ont fait de moi la personne que je suis aujourd'hui. Chacun de vous a apporté une contribution unique à mon parcours, remplissant ma vie de souvenirs précieux et d'amour inconditionnel.

Papi et Mami

À mes chers grands-parents, Je suis reconnaissante pour les moments partagés, les histoires racontées et les enseignements transmis. Votre présence chaleureuse a toujours été un refuge pour moi, un endroit où je me suis sentie aimée et soutenue.

Mère et père

À mes parents : a ma mère ***Malti Aini Nadjiba*** *qui a été une source inépuisable d'amour, de soutien et d'inspiration tout au long de mon parcours. Sa bienveillance, son encouragement constant et sa force indéfectible ont été des piliers essentiels qui m'ont permis d'accomplir ce que je suis aujourd'hui. Je lui suis profondément reconnaissante pour sa présence inconditionnelle et son soutien indéfectible. Ce mémoire lui est dédié avec tout mon amour et ma gratitude infinie.*

et mon père ***Redouane*** *qui m'a inculqué un esprit de combativité et de persévérance et qui m'a toujours poussé et motivé dans mes études. Sans eux,*
certainement je ne sériai pas à ce niveau.

Que dieu, le tout puissant, vous préserve et vous procure santé et longue vie afin que je puisse à mon tour vous combler.

À ma sœur ***Narimene*** *Sa présence, son soutien inébranlable et son esprit combatif ont été une source constante d'inspiration pour moi Je suis reconnaissante d'avoir une sœur aussi exceptionnelle qui a toujours cru en moi et m'a encouragée à poursuivre mes rêves. Et mes frères* ***Mehdi et Merouane*** *, pour ses encouragements incessants.*

A mes très chères collègues de promotion 2023.

À travers ce mémoire, je veux rendre hommage à chaque membre de ma famille. Votre amour, votre soutien et votre influence positive ont été les fondations sur lesquelles j'ai construit mon chemin. Vous m'avez montré l'importance des relations familiales, de l'entraide et de la solidarité.

Que ces mots soient un témoignage de ma gratitude éternelle envers vous, ma chère famille. Votre présence dans ma vie est un cadeau inestimable, et je ne cesserai jamais de vous remercier pour tout ce que vous avez fait et continuez de faire pour moi.

Avec tout mon amour et ma reconnaissance infinie,

Bekara Ines

Table Des Matières

Liste Des Figures

Liste Des Tableaux

Introduction générale

Introduction générale

L'accès à l'information pertinente et précise est un défi majeur dans l'ère de l'explosion des données. Les méthodes traditionnelles de recherche d'information basées sur des mots-clés rencontrent souvent des limites en termes de pertinence. Afin de relever ce défi, le présent mémoire se concentre sur le développement d'un système de recherche d'information à base d'appariement sémantique, qui exploite la signification des concepts plutôt que de se limiter aux termes spécifiques utilisés.

L'objectif principal de ce mémoire est donc de concevoir et de réaliser un système de recherche d'information qui intègre des techniques d'appariement sémantique avancées. Pour atteindre cet objectif, nous adopterons une approche méthodologique rigoureuse, basée sur une analyse approfondie des méthodes d'appariement sémantique existantes et leur adaptation aux besoins spécifiques de notre système.

Dans un premier temps, nous présenterons les concepts clés afin de comprendre la recherche d'information dans son ensemble ainsi nous analyseront les différents modèles utilisés et mettrons en évidence les avantages et les défis rencontres dans ces derniers. Cette introduction permettra de poser les bases nécessaires à la compréhension approfondie de notre recherche.

Par la suite, nous exposant en détails les différentes catégories de mesure de similarité entre phrase. Nous présenterons les avantages et inconvénient de chacune des mesures.

Enfin, nous décrirons en détail notre méthodologie de recherche, en expliquant les choix effectués et les raisons qui les sous-tendent. Nous présenterons les différentes étapes de développement du système de recherche d'information à base d'appariement sémantique, de la collecte des données à la mise en place des techniques d'appariement sémantique. Nous aborderons également les éventuelles limites de notre étude et les stratégies que nous mettrons en place pour les atténuer.

Le manuscrit est organisé en trois chapitres :

- Chapitre 1 : Recherche d'Information

Ce chapitre introduit le domaine de Recherche d'Information et ces concepts fondamentaux.

- Chapitre 2 : Mesure de similarité entre phrase

A travers ce chapitre, nous présentons brièvement une phrase dans la littérature, les catégories des mesures, leurs caractéristiques et fondements de base ainsi que leurs domaines d'application

- Chapitre 3 : Implémentation

Ce dernier chapitre présente l'approche proposée et l'étude de cas qui illustre la mise en œuvre du système.

En conclusion, ce mémoire s'engage à examiner de manière critique les enjeux et les défis de la recherche d'information à base d'appariement sémantique et à proposer des recommandations pratiques et théoriques pertinentes pour l'amélioration des systèmes de recherche d'information.

Chapitre I
Recherche d'Information

Chapitre 1 : Recherche d'Information

1-Introduction :

Internet est devenu un moyen de communication efficace. L'outil informatique nous facilite la création et publications des documents peu importe leurs quantités et formats. Il existe actuellement près de 100 milliards de pages sur la toile [1]. Comment trouver une information ? Comment ne retrouver que l'information recherchée ? Comment trouver l'information qui répondrait exactement le mieux aux besoins d'utilisateur ? Trouver une aiguille dans une botte de foin ! Est-ce possible ? Est-ce raisonnablement faisable ? Les Systèmes de Recherches d'Informations (SRI) rapportent des réponses plus ou moins satisfaisantes à ces questions. Ils permettent de retourner à partir des fonctions de comparaisons et d'un ensemble de documents, ceux dont le contenu correspond le mieux à un besoin d'utilisateur exprimé à l'aide d'une requête.

Dans ce chapitre nous va présenter les concepts de base de la Recherche d'Information (RI). Notre but est de définir les notions document, requête et pertinence. Par la suite décrire le processus et les différents modèles de la RI et on clôturera cette section avec l'évaluation des SRI.

2-Définition de la recherche d'information (RI) :

Il existe plusieurs définitions de la recherche d'information voici quelques-unes :

Définition 1 :« Ensemble des méthodes et techniques pour l'acquisition, l'organisation, le stockage, la recherche et la sélection d'information pertinente pour un utilisateur »[1]

Définition 2 :« IR is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers). » [2]

Définition 3 :« Information retrieval deals with the representation, storage, organization of, and access to information items such as documents, Web pages, online catalogs, structured and semi-structured records, multimedia objects. The representation and organization of the information items should be such as to provide the users with easy access to information of their interest. » [3]

Définition 4 :« IR : The techniques of storing and recovering and often disseminating recorded data especially through the use of a computerized system. » [4]

Plusieurs concepts clé peuvent être définit, on distingue les plus importants dans la RI à savoir :

La requête : constitue l'expression du besoin en information de l'utilisateur. Elle représente l'interface entre le SRI et l'utilisateur **[5],** et peut être exprimé sous forme d'une expression booléenne (AND, OR, NOT), d'autres SRI utilisent une liste de mots clés tandis que d'autres permettent d'introduire en langage naturel.

Les requêtes peuvent être navigationnelles, transactionnelles, informationnelles. La première est quand un utilisateur cherche un site en particulier (exemple : **si quelqu'un tape « netflix » dans Google)**, la seconde lorsque l'utilisateur souhaite acheter quelque chose de spécifique, mais ne sait pas encore où il va se le procurer (exemple : **si quelqu'un tape « acheter tapis de course »**) et la dernière correspond à des recherches d'informations (exemple : **si quelqu'un tape « qu'est-ce qu'une balise HTML »).**

La collection de document : Elle constitue l'ensemble des informations exploitables et accessibles par le Système de Recherche d'Information (SRI). Le document peut être un texte, une page web, une image, une bande vidéo **[6]**. En résumé, un document est une unité qui peut constituer une réponse à un besoin en information exprimé par un utilisateur.

Pertinence : Le but de la RI est de trouver seulement les documents pertinents. La notion de pertinence est très complexe est définit comme :

« In information science and information retrieval, relevance denotes how well a retrieved document or set of documents meets the information need of the user. Relevance may include concerns such as timeliness, authority or novelty of the result» [7]

Les chercheurs et les concepteurs de SRI distinguent deux types de pertinence : la pertinence système, c'est-à-dire l'évaluation par un système de l'adéquation entre des documents et une requête, et la pertinence utilisateur qui se traduit par des jugements de pertinence sur les documents fournis en réponse à une requête. [8]

3- Processus de Recherche d'Information (RI)

Il y a trois étapes fondamentales dans le déroulement du processus : indexation, appariement documents/requêtes et évaluation. Elles sont représentées schématiquement par le processus en U dans la Figure 1.

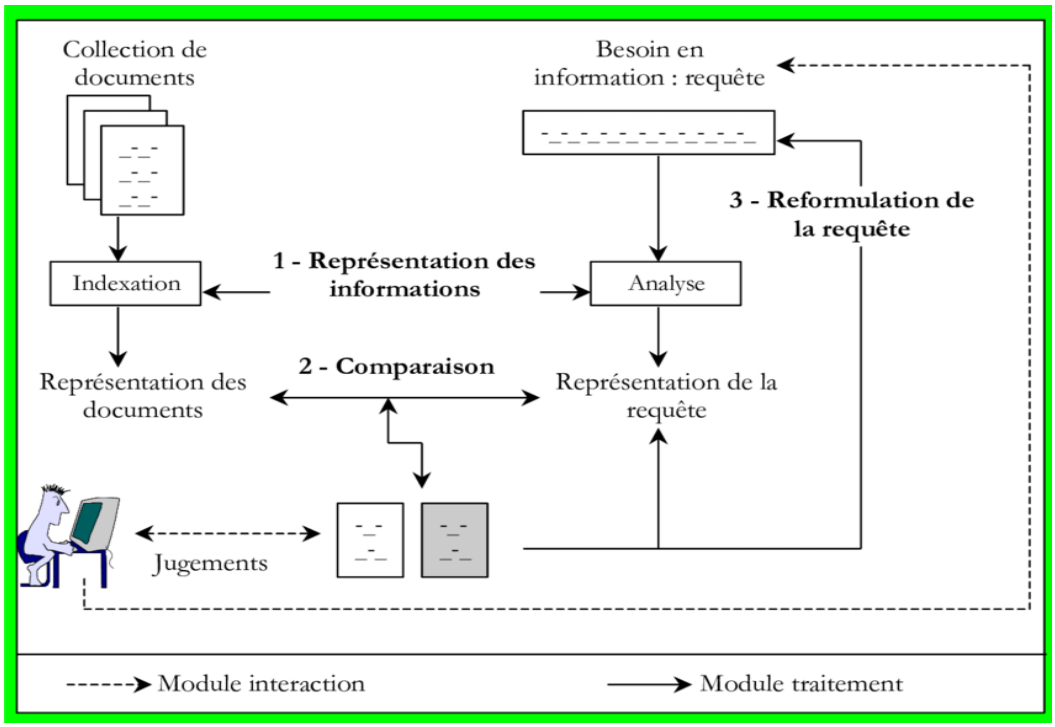


Figure I. 1-Processus en U [9]

3.1. Indexation :

L'indexation a pour rôle de transformer un document textuel ou une requête, en un ensemble de descripteur qui montre le mieux son contenu sémantique. Plusieurs méthodes de représentation existent les plus utilise sont les suivantes :

1. ***Représentation en sac de mots (bag of words) :*** les textes sont transformés simplement en mots (termes).
2. ***Représentation avec les racines lexicales*** : remplacer les mots du document par leurs racines lexicales, et à regrouper les mots de la même racine dans une seule composante.[10]
3. ***Représentation avec les lemmes*** : La lemmatisation consiste à utiliser l'analyse grammaticale afin de remplacer les verbes par leur forme infinitive et les noms par leur forme au singulier. [10]

4. ***Représentation avec les n-gramme*** : Elle consiste a découpé le texte en plusieurs séquence de n caractère.
5. ***Représentation par phrases*** : utiliser les phrases comme unité de représentation.
6. ***Représentation conceptuelle*** : représenter le document sous forme d'un ensemble de concept.

Tableau I. 1-Avantages et Inconvénient des méthodes de représentation

Type de descripteurs	Avantages	Inconvénients
Représentation en sac de mots (bag of words)	▪ Exclure toute analyse grammaticale ▪ Exclure la notion de distance entre les mots	▪ Les mots composés allemands peuvent être très complexes, ▪ Le chinois et le japonais ne séparent pas les mots par des espaces, ce qui peut mener à plusieurs segmentations, ▪ L'arabe et l'hébreu sont écrits de droite à gauche, mais certains éléments tels que les nombres sont écrits de gauche à droite.
Représentation avec les racines lexicales	▪ Diminue la taille d'un document.	▪ Une racine commune pour des mots qui portent des sens différents. ▪ Diffère d'une langue à autres.
Représentation avec les lemmes	▪ Elle est simple. ▪ Réduire la dimension.	▪ Perdre la sémantique. ▪ Considère les synonymes comme des lemmes différents.
Représentation avec les n-gramme	▪ C'est une méthode indépendante de la langue. ▪ Capture les racines des mots les plus fréquents.	▪ Perdre la sémantique.
Représentation par phrases	▪ Plus informatives que les mots seuls. ▪ Conserve l'information relative à la position du	▪ Le nombre important de phrases.

	mot dans la phrase.	
Représentation conceptuelle	▪ Réduire l'espace de représentation car les mots qui sont synonymes partagent le même concept.	▪ Il n'existe pas des bases lexicales pour toutes les langues.

On distingue trois modes d'indexation **manuelle, automatique et semi-automatique.** La première est réalisée par un documentaliste (spécialiste), Son résultat est précis et exacte mais elle est couteuse en termes de temps et d'espace. La deuxième est complètement informatisée basée sur un programme. La dernière commence par indexer de façon automatique avec l'aide d'un documentaliste (spécialiste du domaine) pour ajouter des relations sémantiques entre les mots.

L'indexation automatique est la plus utilisé, ses étapes sont montrées dans la figure 2 suivante.

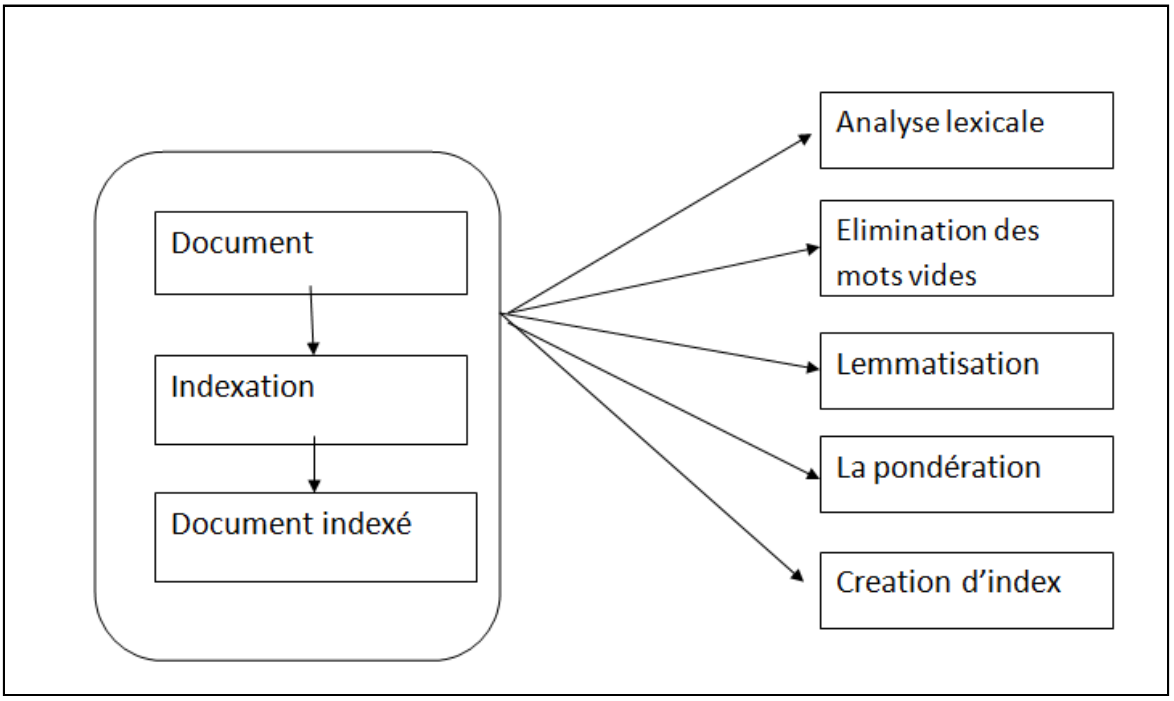


Figure I. 2-étapes d'indexation automatique

- ***L'analyse lexicale (Tokenisation)***

C'est l'étape importante pour l'identification des unités lexicales du texte car elle consiste à convertir une chaine de caractères (le texte) en une séquence d'unités lexicales essentiels appelées « Token », ces dernières sont candidates à l'indexation. Cependant la tokenisation nécessite La conversion de la casse (transformer les majuscules en minuscule), l'élimination des accents la tokenisation à l'aide d'un algorithme.

- ***L'élimination des mots vides***

Un des problèmes majeurs de l'indexation consiste à éviter les mots vides qui sont des mots athématiques (les mots qui peuvent se retrouver dans n'importe quel document parce qu'ils exposent le sujet mais ne le traitent pas, comme par exemple contenir, appartenir, etc.) [1]

Pour les filtrer on peut utiliser une liste prédéfinie de mots vides (aussi appelée anti-dictionnaire ou stop-list) et aussi L'élimination des mots dépassant un certain nombre d'occurrences. Les deux méthodes sont efficaces mais chacune a des inconvénients. La première diffère de langues à une autre langue et la seconde supprime des mots qui dépassent un certain seuil mais ils sont porteurs de sens (pertinent).

- ***La lemmatisation :***

La lemmatisation est une technique couramment utilisée pour normaliser les formes fléchies des mots dans un texte. Cette méthode permet de regrouper les mots de la même catégorie grammaticale et les transformer en leur forme canonique appelée lemme (exemple : les différentes formes d'un verbe sont transformées à son infinitif) [10]. Cette technique facilite la comparaison entre les mots et améliore la précision des analyses textuelles et permet de mieux préserver le sens des mots dans le texte et peut donc améliorer la qualité des résultats mais elle cause une perte d'information et peut être complexe, car elle nécessite des règles grammaticales et des dictionnaires pour identifier les lemmes corrects, ce qui peut rendre le processus de traitement plus long et plus coûteux.

- ***La pondération***

Son principe est d'affecter à chaque terme **t** d'un document **d** ou d'une requête **q**, un poids numérique qui le caractérise dans le document ou la requête, les poids des termes de la

requête et du document peuvent avoir des sémantiques différentes, alors le poids est une mesure statistique de l'importance du terme dans le document (plus un terme est important dans le document plus son poids dans ce document doit être élevé). Le tableau illustre les méthodes de pondération utilisées :

Tableau I. 2-Différentes méthodes de pondération

Mesure	***Formule***
TF	Fréquence du terme dans le document.
TF Normalisé	Fréquence du terme dans le document/ la taille du document.
IDF	Mesure l'importance d'un terme dans toute la collection. **Log(N/df) où** ▪ N : nombre totale de doc dans la base. ▪ Df : nombre de documents contenant le terme.
TFIDF	$\boldsymbol{tft,d * idft}$
TFC	$TFC(t_k, d) = \frac{TF \times IDF(t_k, d)}{\sqrt{\sum^{|r|} (TF \times IDF(t_s, d))^2}}$

- ***Création d'index :***

Dans la recherche d'information, un index est une structure de données utilisée pour stocker des informations sur les termes qui apparaissent dans un corpus de documents. L'index permet de retrouver rapidement les documents qui contiennent un terme donné.

Il existe plusieurs méthodes pour créer un index parmi eux on a le fichier inverse (ou index inversé) qui est une structure de données utilisée pour permettre la recherche de mots-clés dans un corpus de documents. Dans un fichier inverse, chaque terme présent dans les documents est associé à la liste des documents qui le contiennent. La structure est la suivante :

Mot -> liste de <documents contenant le mot et sa fréquence>

3.2. L'appariement document-requête

L'appariement entre une requête et un document est primordiale dans la RI. C'est un processus de correspondance entre les termes d'une requête de recherche et les termes d'un document pertinent dans une collection de documents. L'objectif de cet appariement est de trouver les documents les plus pertinents pour une requête donnée selon une mesure de similarité. Cette dernière est calculée à partir d'une fonction intitulée RSV (Q, D) (Retrieval Status Value), où Q est la requête et D un document [11]. On différencie deux types d'appariement « exacte » et « approché » :[12]

- Appariement exact : Le résultat est une liste de documents respectant exactement la requête spécifiée avec des critères précis. Les documents retournés ne sont pas triés.
- Appariement approché : Le résultat est une liste de documents censés être pertinents pour la requête. Les documents retournés sont triés selon un ordre de mesure. Cet ordre reflète le degré de pertinence document/requête.

4- Les modèles de Recherches d'Informations :

Un modèle de Recherche d'Information propose une manière unifiée de représenter les requêtes et les documents ainsi qu'une fonction de correspondance (pertinence) qui associe des scores aux couples requête-document permettant ainsi de trier les documents en fonction de la requête.[13]

Une taxonomie des modèles a été présentée par Baeza-Yates and Ribeiro-Neto (1999) [14] qui présente quatre familles principales. Les **modèles de RI classiques, modèles basés sur le texte semi-structuré**, **modèles orientés web** et **la recherche d'images, de musiques, d'audio ou de vidéos**. On s'intéresse aux modèles de RI classique illustrée dans la Figure 3.

Selon Baeza-Yates and Ribeiro-Neto [14] : Un modèle de RI est défini par un quadruplet (D, Q, F, R (q, d)) où : D est l'ensemble de documents ; Q est l'ensemble de requêtes ; F est le schéma du modèle théorique de représentation des documents et des requêtes ; R (q, d) est la fonction de pertinence du document d à la requête q.

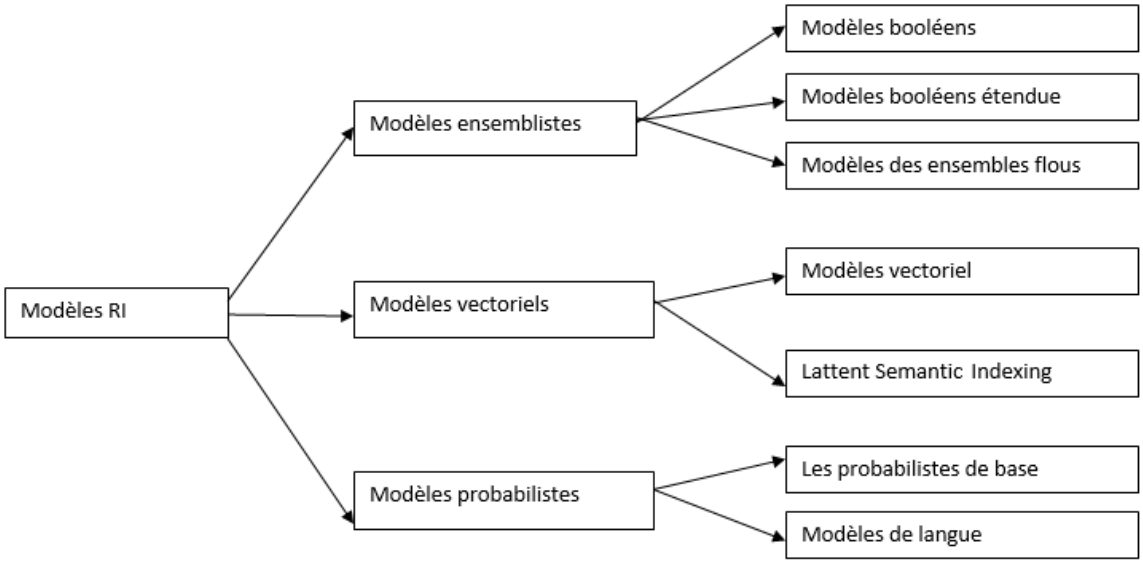


Figure I. 3.Les Modèles de Recherche d'informations.

4.1. Le modèle ensembliste

Ce sont les modèles qui sont basés sur la théorie des ensembles et l'algèbre de bool. Le modèle booléen est le premier modèle utilisé dans le domaine de la RI même aujourd'hui beaucoup de systèmes commerciaux (moteurs de recherche) utilisent le modèle booléen. Cela est dû à la simplicité et à la rapidité de sa mise en œuvre [15]. Dans ce modèle, le document est représenté par un ensemble de termes. Par contre la requête est représentée sous forme d'une expression logique formée de termes reliés par des opérateurs booléens AND, OR et NOT [16]. L'appariement se base sur **RSV (d,q) (RetrievalStatut Value)** : fonction de correspondance entre document d et différentes forme de requêtes q.

RSV (d,q)={ 1 ou 0 }

La simplicité du modèle le rend plus compréhensible pour un utilisateur mais représente des faiblesses tel que :[17]

- La sélection d'un document est basée sur une décision binaire.
- Pas d'ordre pour les documents sélectionnés.
- Formulation de la requête difficile et pas toujours évidente pour beaucoup d'utilisateurs.
- Problème de collections volumineuses : le nombre de documents retournés peut être considérable.

Et pour résoudre ces problèmes, des extensions ont été proposées le modèle des ensembles flous, le modèle booléen étendu [19]. Le tableau suivant nous les explique :

Tableau I. 3 les extensions du modèles booléens

Modèles	Fonctionnement	Avantages	Inconvénients
Modèle booléen étendu	Combinaison des modèles booléen et vectoriel	-Prendre en compte l'importante des termes dans les documents et/ou dans la requête	Calcul complexe Problème de distributivité
Modèles des ensembles flous	Intègre le principe des pondérations	-Possibilité d'ordonner les documents sélectionnés	

4.2. Le modèle vectoriel

C'est le plus populaire en RI introduit par Salton [19]. Les requêtes et les documents sont alors représentés par des vecteurs, dont les composantes représentent le poids du terme d'indexation considéré dans le document (la requête). Formellement, si on a un espace T de termes d'indexation de dimension N.

T = {t1, t2, …tj, …tn}.

Un document di est représenté par un vecteur.

di (wi1, wi2,.....,wij,.....win).

Une requête q par un vecteur.

q (wq1, wq2,..... wqj,....wqn).

Où wij (resp. wqj) représente le poids du terme tj dans le document di (respectivement dans la requête q). Ce poids peut être soit une forme de $\boldsymbol{tft}$, $*\boldsymbol{idft}$, soit un poids attribué manuellement par l'utilisateur.[15]

La pertinence du document di vis-à-vis de la requête Q est mesurée comme le degré de corrélation des vecteurs correspondants. Cette corrélation peut être exprimée par l'une des mesures suivantes illustrées dans le tableau suivant : [20]

Tableau I. 4. *Les mesures de similarité utilisées dans le modèle vectoriel [21]*

Mesures	Formules
Le produit scalaire	$RSV(q,d_i) = \sum_{j=1}^{\|T\|} w_{qj}.w_{ij}$
La mesure de cosinus	$RSV(q,d_i) = \frac{q.d_i}{\|q\|.\|d_i\|} = \frac{\sum_{j=1}^{\|T\|} w_{qj}.w_{ij}}{\sqrt{\sum_{j=1}^{\|T\|} {w_{qj}}^2 \sum_{j=1}^{\|T\|} {w_{ij}}^2}}$
La mesure de Dice	$RSV(q,d_i) = \frac{2 \times \sum_{j=1}^{\|T\|} w_{qj}.w_{ij}}{\sum_{j=1}^{\|T\|} {w_{qj}}^2 + \sum_{j=1}^{\|T\|} {w_{ij}}^2}$
La mesure de Jaccard	$RSV(q,d_i) = \frac{\sum_{j=1}^{\|T\|} w_{qj}.w_{ij}}{\sum_{j=1}^{\|T\|} {w_{qj}}^2 + \sum_{j=1}^{\|T\|} {w_{ij}}^2 - \sum_{j=1}^{\|T\|} w_{qj}.w_{ij}}$

En plus de bénéficier d'une mise en œuvre facile, L'avantage du modèle vectoriel est que La pondération améliore les résultats de recherche ; La mesure de similarité permet d'ordonner les documents selon leur pertinence vis à vis de la requête. Cependant, il présente l'inconvénient de reposer sur l'hypothèse d'indépendance des termes –bag of words– alors que ce sont parfois les expressions ou les groupes de mots qui enrichissent la sémantique du document. Une solution est proposée par le modèle d'indexation sémantique latente qui propose d'utiliser des techniques d'analyse multidimensionnelle des termes.

4.3. Le modèle probabiliste

Ce modèle est fondé sur le calcul de la probabilité de pertinence d'un document d pour une requête q. On se rapproche ici de la notion de classification probabiliste. L'idée est de retrouver des documents qui ont en même temps une forte probabilité d'être pertinents, et une faible probabilité d'être non pertinents [22]. Un document est sélectionné si la probabilité qu'il soit pertinent pour Q notée P(R|d) est supérieure à la probabilité qu'il soit non pertinent pour Q, notée P(NR|d).

RSV (d,Q) est donné par :

$$RSV\ (d,Q)=\frac{P(R|\ d)}{P_{(NR|d)}}$$

P(d|R) (respectivement P(d|NR)) est la probabilité que le document appartienne à l'ensemble l'ensemble R des documents pertinents (respectivement à l'ensemble NR des documents non pertinents).

Le modèle probabiliste de base est fonctionnel et efficace. Cependant, son inconvénient est de considérer les termes d'indexation comme étant indépendants les uns des autres, ce qui n'est généralement pas le cas.

5-La reformulation de requête

Ceux qui utilisent les moteurs de recherche ne sont pas des professionnels de la documentation. Ainsi les utilisateurs ne savent pas choisir les bons termes (mot-clé) pour exprimer ses besoins en information. En introduisant la reformulation de requête, la RI est alors envisagée comme une suite de formulations et de reformulations de requêtes jusqu'à la satisfaction du besoin en information d'un utilisateur. Toujours la requête initiale aboutie rarement à un résultat qui satisfait ce dernier. Il s'agit donc d'ajouter des termes à la requête initiale de l'utilisateur et on parle alors d'expansion de la requête de l'utilisateur. On distingue trois types de reformulation :

5.1. La reformulation manuelle :

C'est une approche utilisée pour les systèmes de recherches booléens car elle est complexe. On procède à la reformulation de requête en utilisant un vocabulaire qui est contrôlé par exemple les thesaurus ou classification pour permettre à l'utilisateur de trouver les bons termes pour compléter sa requête.

5.2. La reformulation automatique :

On parle de reformulation automatique, Lorsque le feedback de pertinence s'accompagne d'ajout (et/ou) de suppression de termes. Les demandes des utilisateurs sont automatiquement retraitées pour intégrer les descripteurs des documents jugés pertinents ou rejetés. Il existe différentes variantes de cette technique : celles qui reformulent automatiquement la requête en augmentant le poids des termes présents dans les documents jugés pertinents, et inversement pour diminuer le poids des termes jugés non pertinents.

Le problème avec la reformulation automatique est l'estimation du terme "bon", qui peut en fait conduire à une amélioration du processus de recherche, car l'introduction de termes inappropriés conduit à du silence ou du bruit.

5.3. La reformulation interactive :

Dans la reformulation interactive, l'utilisateur est actif. A l'inverse de la reformulation automatique, le système et l'utilisateur sont, ensemble, responsables de la détermination et du choix des termes candidats à la reformulation. Le système joue un rôle dans la suggestion des termes, le calcul des poids des termes et l'affichage à l'écran de la liste ordonnée des termes. L'utilisateur examine cette liste et décide du choix des termes à ajouter dans la requête. C'est donc l'utilisateur qui prend la décision finale dans la sélection des termes.

6- Evaluation des systèmes de recherches d'informations

L'évaluation d'un SRI permet de vérifier l'efficacité des modèles mis en œuvre pour l'identification des documents pertinents. Elle permet également de paramétrer le modèle, d'estimer l'impact de chacune de ses caractéristiques et de fournir des éléments de comparaison entre modèles. Cleverdon en 1970 avait défini les collections de tests ainsi que des mesures d'évaluations. D'une manière générale, pour évaluer un SRI on doit avoir une collection de test, ainsi on procède de la façon suivante : le système exécute les requêtes une par une sur la collection de documents et renvoie pour chacune une liste ordonnée de documents qu'il considère comme potentiellement pertinents.[23] Chaque document est ensuite comparé aux jugements de pertinence et des mesures d'efficacité sont calculées. Nous présentons ces mesures et la collections de tests (documents) dans cette section.

6.1-les mesures d'évaluations :

Il y a plusieurs mesures de la qualité d'un SRI, dont : le temps de réponse, la présentation des résultats, l'effort requis de l'utilisateur pour retrouver parmi les documents retournés ceux qui répondent à son besoin autrement dit la pertinence qui est évaluée grâce aux deux facteurs : **le taux de rappel du système** et **la précision du système.** On les présente ci-dessous :

- Rappel : est le rapport de documents pertinents restitués par le système sur l'ensemble des documents pertinents contenus dans la base documentaire. Elle mesure la capacité du système de retrouver tous les documents pertinents répondant à la requête.

$$rappel = \frac{\text{nombre de documents pertinents trouvées}}{\text{nombre de documents pertinents}}$$

- Précision : est le rapport de documents pertinents trouvés sur l'ensemble des documents restitués par le système. Elle mesure la capacité du système à rejeter tous les documents non pertinents à une requête donnée.

$$precision = \frac{\text{nombre de documents pertinents trouvées}}{\text{nombre de documents}}$$

Idéalement, on souhaiterait qu'un système donne de bons taux de précision et de rappel en même temps. Un système qui aurait 100% pour la précision et pour le rappel signifie qu'il trouve tous les documents pertinents, et rien que les documents pertinents. Cette situation n'arrive pas. Plus souvent, on peut obtenir un taux de précision et de rappel aux alentours de 30%.

Les valeurs de ces mesures sont comprises entre 0 et 1 et sont optimales pour 1. Les deux métriques sont dépendantes. Il y a une forte relation entre elles. Ces deux mesures varient en sens inverse : les méthodes permettant d'augmenter la précision ont tendance à dégrader le rappel et vice versa. La mesure F permet de combiner le rappel et la précision comme suit :

$$F = \frac{2PR}{P + R}$$

Cette mesure donne la même importance à la précision et au rappel. Des variantes (Fβ) permettent de donner plus d'importance à l'un ou à l'autre.

$$F\beta = \frac{(1 + \beta^2)PR}{\beta^2 P + R}$$

β contrôle le compromis R, P:

– β = 1: même poids précision et rappel (Fβ=F).

– β > 1: privilégie la précision au rappel

– β < 1: plus d'importance au rappel.

On retrouve aussi La précision moyenne (AP pour Average Precision) est également largement utilisée pour évaluer un système ou pour comparer des systèmes. On a Deux façons de calculer la moyenne : Micro-moyenne – chaque document pertinent est un point de la moyenne et Macro-moyenne – faire la moyenne par requête.

Toutes ces mesures sont calculées pour une requête donnée. Cependant, les moteurs doivent être évalués sur un ensemble de requêtes ; les mesures précédentes se déclinent alors en valeur moyenne. Par exemple, la moyenne de la précision moyenne (MAP pour Mean Average Precision) est la moyenne pour un ensemble de requêtes de l'AP obtenue pour chaque requête [24].

Le rappel et la précision permettent aussi de définir le silence documentaire et le bruit documentaire qui représentent respectivement le nombre des documents pertinents non retrouvés et le nombre de documents non pertinents retrouvés comme illustre la figure ci-dessous.

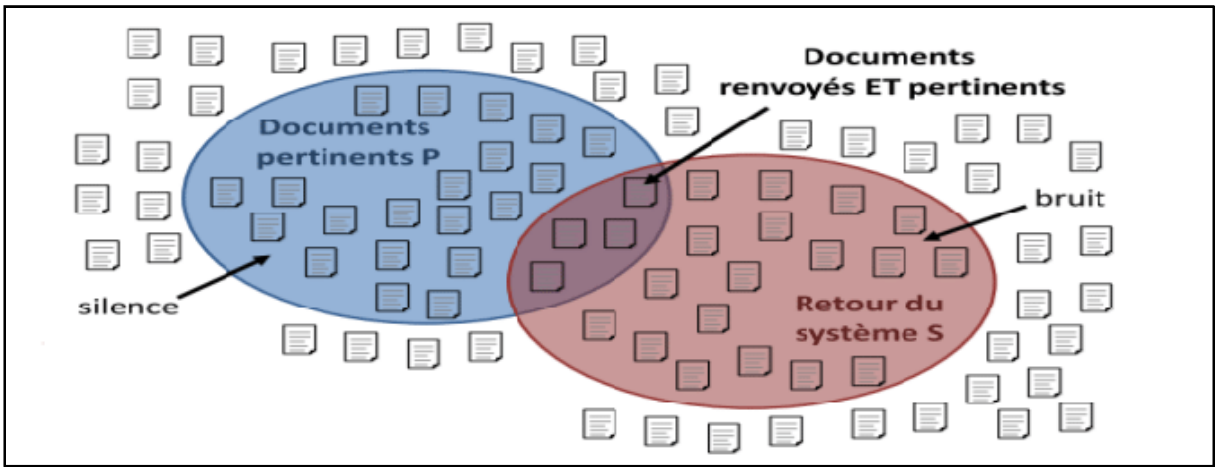


Figure I. 4.Schéma d'évaluations des SRI

6.2-la collections de tests :

La collection de test (ou corpus) constitue la méthode d'évaluation du SRI. Il se compose d'un ensemble de documents à indexer sur le système à évaluer, d'une liste de requêtes prédéfinies, et de jugements de pertinence, construits manuellement pour chaque requête (la liste des documents jugés pertinents pour cette requête). L'évaluation du SRI consiste à confronter les résultats rendus par ce dernier à des jugements de pertinence. L'ensemble de test est le résultat d'une multiplication d'items d'évaluation depuis les années 1970. On peut citer quelques collections de test dans le tableau suivant :

Tableau I. 5. Type de collections de test

Collection nom	Domaine	Nombre de documents
Adi	Domaine des sciences de l'information.	82
Cacm	Domaine de l'informatique.	3204
Cisi	Domaine des sciences de l'information	1460
Méline	Domaine médical.	1033
Time	Articles du magazine Time.	425

Différentes collections de test sont utilisées en recherche d'information. Parmi elles TREC et CLEF, dans ce qui suit nous détaillerons la collection TREC :

Dans la recherche d'information (RI) différentes collections de test sont utilisées. Parmi elles on a TREC (Text Retrieval Conference,[25]et CLEF (Conference and Labs of the Evaluation Forum,[26]) ou encore NTCIR (NII Test Collection for IR Systems,[27]). Dans ce qui suit nous détaillerons la collection TREC :

« Le **Text REtrieval Conference** (TREC) est un programme conçu comme une série d'ateliers dans le domaine de la Recherche d'information (RI ou IR). Ce programme est soutenu conjointement par le National Institute of Standards and Technology (NIST) et par l'Advanced Research and Development Activity (ARDA) Center du Département de la Défense des États-Unis. Il a débuté en 1992 dans le cadre du projet TIPSTER. Son but est d'encourager les travaux dans le domaine de la recherche d'information en fournissant l'infrastructure nécessaire à une évaluation objective à grande échelle des méthodologies de recherche textuelle et accroitre la rapidité du transfert de technologie »[28]

7-Domaines d'Applications de Recherche d'Information :

On retrouve la RI dans diffère domaines :

– Internet (Web, Forum/Blog search, news) .

– Entreprises (entreprise search).

– Bibliothèques numériques «digital library».

– Domaine spécialisé (médecine, droit, littérature, chimie,mathématique, brevets, software...).
– Nos propres PC (Yahoo! Desktop search).

8- Conclusion :

Dans ce chapitre, nous avons introduit les principales notions et concepts de la recherche d'information (RI). Nous avons donné la définition de la RI et en particulier nous avons exploré les concepts fondamentaux de l'IR à savoir les documents et collections de documents, les requêtes, la pertinence et les besoins d'information. Nous avons développé aussi les processus de la RI tels que l'appariement document-requête, Puis nous avons étudié les différents modèles de recherche et enfin on a examiné les méthodes d'évaluation des systèmes de recherche d'informations.

Chapitre II :

Mesure de similarité entre phrase

Chapitre 2 : Mesure de similarité entre phrase.

1-Introduction

Le concept de similarité est fondamental dans de nombreux domaines de l'informatique et des sciences pour prendre des décisions ou d'extraire des informations utiles. L'un des défis majeurs de l'analyse de texte est de quantifier la similitude entre les phrases. Les mesures de similarité entre phrases sont des outils mathématiques qui permettent de comparer deux ou plusieurs phrases pour déterminer leur degré de similitude. Ces mesures peuvent être basées sur des approches sémantiques ou syntaxiques.

Ce chapitre présente tout d'abord les concepts linguistiques fondamentaux qui sont essentiels à la compréhension d'une phrase. Il inclut également les différentes mesures de similarité entre phrase ainsi que les domaines d'applications pour ces derniers.

2-La phrase dans la littérature

Il existe de nombreuses observations en linguistique qui ont été appliquées à la recherche et à la compréhension des significations des phrases.

- ***Mots (words)*** : Il faut d'abord comprendre les mots pour comprendre les phrases. Lorsqu'on discute des mots, la nomenclature (la classification) suivante sera adoptée :
  - **Forme** : la forme d'un mot est la façon dont il est écrit.
  - **Sens** : l'idée à laquelle le mot se réfère.
  - **Racine** : le mot de base auquel un suffixe a été intégré pour donner la forme.
  - **Suffixe** : un morphème qui a été ajouté à la fin de la racine.
  - **Type** : la "partie du discours" à laquelle appartient une racine.
- ***Phrase (sentence)*** :

  « Une phrase dans la littérature est définie comme une unité grammaticale constituée d'un groupe de mots organisés de manière cohérente et expressive pour transmettre une idée ou un sens complet. » [44]

Afin de discuter de la similarité des phrases, certains termes doivent être définis :

- **Sujet (Topic)** : Le sujet d'une phrase est celui dont parle la phrase. Les phrases peuvent avoir des significations différentes. Et partage un sujet commun. Il s'agit des informations stockées dans un index. Dans un index, un mot-clé sert à vous orienter vers les informations liées à ce mot-clé. Toutefois, il ne fournit pas d'indication sur le contenu spécifique de ces informations.
- **Signification (Meaning)** - Cela décrit l'idée que la phrase transmet dans son ensemble. Il ne s'agit pas seulement de ce dont parle la phrase, mais de l'idée et de l'action à laquelle la phrase fait référence.
- **Interaction entre les mots (Word Interaction)** - Comment les mots se combinent pour donner le sens de la phrase au-delà des significations de ses mots et donc au-delà du sujet.
- **Contexte (Context)** - Comment les significations des mots sont fixées à partir des mots environnants. Cela peut être très complexe car cela dépend également des idées discutées.

En résumé les trois aspects (topic,words interactions et context), sont cruciaux pour déterminer la signification des mots et des phrases. Car pour mesurer la similarité entre les phrases il est nécessaire de savoir qui est le sujet, comment les mots interagissent entre eux et dans quel contexte.

- **Structure d'une phrase :** selon McArthur [29], il est pratique de considérer la structure qui comprend une clause sujet, une clause verbe et une clause objet comme montré dans la figure :

Figure II. 1. Structure d'une phrase

À titre d'exemple, une phrase simple est divisée par {} pour indiquer les clauses :

{The man} {killed} {the thief}.

"killed" est la clause verbale qui décrit l'action de "kill" et qui signifie que c'est un événement qui s'est déjà terminé en utilisant le temps passé. Elle est essentielle pour permettre la fonction d'une clause transformationnelle et décrit l'action qui se déroule ainsi que le changement temporel par rapport à l'orateur de l'action (passé, présent ou futur).

"The man" est la clause sujet car c'est lui qui effectue l'action.

"The thief" est la clause objet car l'action a été effectué sur le voleur.

Le sujet et l'objet sont distingués par leurs positions relatives par rapport à la clause verbale et bien que l'ordre sujet-verbe-objet (SVO) soit le format normal en anglais et pourrait être décrit comme une langue SVO, il est parfaitement valable de voir l'ordre des clauses altérées.

Lorsque l'ordre SVO est modifié, cela peut être fait en utilisant une voix passive pour que le sujet de la phrase fonctionne comme objet, ou en changeant l'ordre des clauses pour mettre l'accent sur un élément particulier. Ce changement d'ordre nécessite une pause dans la parole pour permettre à l'auditeur de savoir que l'ordre a été modifié et en grammaire écrite, cette pause sera normalement indiquée par la ponctuation.

- Type de phrase : Les types de phrase sont caractérisés par une syntaxe, une morphologie et une intonation qui leur sont propres. On les reconnait donc à différentes marques.[30]
  - La phrase déclarative (ou *assertive*) : sert à déclarer ou à affirmer quelque chose. C'est le type de phrase le plus fréquent, et c'est SVO.
  - La phrase interrogative : on construit une phrase interrogative Lorsqu'on pose directement une question, qu'on interroge quelqu'un.
  - La phrase impérative (ou *injonctive*) :sert à ordonner quelque chose à quelqu'un. Cet ordre peut en fait être un conseil, une recommandation, un souhait, une prière ou une invitation. Dans tous les cas, on incite la personne a qui l'on s'adresse à agir. On reconnaît la phrase impérative à deux particularités : le verbe est à l'impératif et il n'y a pas de sujet exprimé.
  - La phrase exclamative permet d'exprimer des sentiments ou des jugements avec intensité. Elle se caractérise par un mot

exclamatif (déterminant ou adverbe) en début de phrase et par un point d'exclamation comme ponctuation finale.

Tableau 5. Exemple des types et leurs formes [31].

	Phrase déclarative	Phrase interrogative	Phrase exclamative	Phrase impérative
Forme affirmative	Le chat est sorti.	Le chat est-il sorti ?	Oh ! Le chat est sorti !	Fais sortir le chat !
Forme négative	Le chat n'est pas sorti.	Le chat n'est-il pas sorti ?	Oh ! Le chat n'est pas sorti !	Ne fais pas sortir le chat !

3-Mesures de similarités entre phrases :

Dans la recherche d'information, la mesure de similarité est utilisée pour attribuer un score de classement entre une requête et un corpus. Il existe trois classes de mesures qui peuvent être utilisées pour identifier la similarité entre des phrases. La première classe est Word Overlap Measures (Mesures de chevauchement de mots) qui prend en compte l'emplacement des mots dans les phrases. La seconde est TF-IDF Measures (Mesures TF-IDF) qui utilise l'importance des mots (le poids) dans les phrases.La dernière est Linguistic Measures (Mesures linguistiques)) prend en compte les relations sémantiques pour les calculs.

3.1. Word Overlap Measures (Mesures de chevauchement de mots) :

Elle calcule le score de similarité en fonction du nombre de mots partagés par deux phrases. On retrouve quatre mesures de chevauchement de mots : le coefficient de similarité de Jaccard, le chevauchement de mots simple, le chevauchement IDF et le chevauchement de phrases.

1- Jaccard coefficient (Coefficient de Jaccard)

Le coefficient de Jaccard est une mesure de similarité qui compare deux ensembles en calculant la proportion de leur intersection par rapport à leur union. Dans le contexte du traitement de texte, elle est souvent utilisée pour mesurer la similarité entre deux ensembles de mots, tels que ceux qui apparaissent dans deux phrases différentes.

La formule suivante est utilisée pour calculer la similarité :

$$J(S1,S2) = \frac{|S1 \cap S2|}{|S1 \cup S2|}$$

Où S1 et S2 sont les deux phrases, $|S1 \cap S2|$ est la cardinalité de l'intersection de S1 et S2, et

$|S1 \cup S2|$ est la cardinalité de l'union de S1 et S2.

Lorsque le coefficient de Jaccard tend vers 1, cela implique que les deux phrases sont étroitement liées et présentent une grande similarité. À l'inverse, lorsqu'il tend vers 0, cela suggère que les deux phrases sont largement distinctes et ont une faible similarité. [32]

Exemple :

Supposons que nous avons les deux phrases suivantes :

Phrase 1 : "Le chat noir dort sur le tapis" Phrase 2 : "Le chien brun joue dans le jardin"

Phrase 1 : {chat, noir, dort, sur, tapis} Phrase 2 : {chien, brun, joue, dans, jardin}

Intersection : {} (il n'y a aucun mot commun entre les deux phrases).

Union : {chat, noir, dort, sur, tapis, chien, brun, joue, dans, jardin}

Coefficient de Jaccard :

$$J(\text{phrase1}, \text{phrase2}) = 0 / 10 = 0$$

Dans cet exemple, le coefficient de Jaccard entre les deux phrases est de 0, ce qui indique qu'il n'y a pas de similarité entre les deux phrases.

2- Simple word overlap (chevauchement de mots simple)

La mesure de similarité de Simple Word Overlap (SWO) est définie comme la proportion de mots présents dans les deux phrases, normalisée par la longueur de chaque phrase.[33]

La formule suivante est utilisée pour calculer la similarité :

$$\text{Sim\_swo}(S1, S2) = \frac{|S1 \cap S2|}{\max(\text{len}(S1), \text{len}(S2))}$$

Où S1 et S2 sont les deux phrases, $|S1 \cap S2|$ est l'ensemble de mots qui se trouve dans S1 et S2, len(S1) est la taille de la phrases S1 (resp. S2) [33]**.**

Le principe de SWO est basé sur l'intuition que deux phrases qui partagent de nombreux mots en commun sont plus susceptibles de parler du même sujet ou d'exprimer des idées similaires.

Par exemple, on a les deux phrases suivantes :

Phrase 1 : "Le chat noir chasse la souris." Phrase 2 : "Le chien brun court après le chat noir."

Phrase 1 {"Le", "chat", "noir", "chasse", "la", "souris"}.

Phrase 2 {"Le", "chien", "brun", "court", "après", "le", "chat", "noir"}.

Mots en communs :("Le", "chat" et "noir").

La similarité de SWO :

$$\text{Sim (phrase1, phrase2)} = 3/8 = 0.37$$

Donc les deux phrases sont peu similaires.

C'est une méthode simple mais souvent efficace pour mesurer la similarité entre deux phrases. Cependant, elle ne prend pas en compte la sémantique des mots ou la structure syntaxique des phrases, ce qui peut être limitatif dans certains cas.

3- Idf chevauchement (le chevauchement IDF**)**

L'IDF Overlap Measures (mesure de chevauchement IDF) est une méthode de calcul de similarité entre deux phrases qui prend en compte la fréquence des mots dans une phrase. Contrairement à la mesure de similarité de SWO qui ne prend en compte que la présence ou l'absence des mots, cette mesure utilise la fréquence inverse de document (IDF) pour pondérer les mots en fonction de leur importance [33].

La formule est :

$$\text{Sim\_idf}\ (S1, S2) = \frac{\sum(\text{idf}(w) * \min(\text{tf}(w, S1), \text{tf}(w, S2)))}{\sqrt{(\sum(\text{idf}(w)\text{^}2))}}$$

Où idf(w) est l'inverse document frequency du mot w dans la collection de documents, tf(w, S1) est la fréquence du mot w dans la phrase1 (c'est-à-dire le nombre de fois où le mot w apparaît dans la phrase1) resp.(tf(w,S2)).

Le principe est que les mots qui sont fréquents ont une faible importance pour la mesure de similarité, tandis que les mots moins fréquents ont une importance plus élevée.

Considérons les deux phrases :

Phrase 1 : "Le chat noir dort sur le tapis" Phrase 2 : "Le chien brun joue dans le jardin"

Supposons qu'on a 100 phrases et l'IDF est calculé pour chaque mot en fonction des phrases.

IDF={IDF("Le") = 0.5 ; IDF("chat") = 1.2 ; IDF("noir") = 1.8 ; IDF("dort") = 2.3 ; IDF("sur") = 0.9 ; IDF("le") = 0.5 ; IDF("tapis") = 2.1 ; IDF("chien") = 1.6 ; IDF("brun") = 1.5 ; IDF("joue") = 2.2 ; IDF("dans") = 1.4 ; IDF("jardin") = 2.0}.

Calcule de idf_overlap : {IDF("Le") * min (tf("Le", phrase1), tf("Le", phrase2)) = 0.5 * 1 = 0.5 ;IDF("chat") * min (tf("chat", phrase1), tf("chat", phrase2)) = 1.2 * 0 = 0}

√(∑(idf(w)^2)) = √((0.5^2) + (1.2^2) + (1.8^2) + (2.3^2) + (0.9^2) + (0.5^2) + (2.1^2) + (1.6^2) + (1.5^2) + (2.2^2) + (1.4^2) + (2.0^2))= 5.64.

IDF overlap :

Sim_idf (S1,S2)=2,6/5,64= 0.461

4. Phrasal overlap (chevauchement inter-phrase) :

Selon Banerjee et Pedersen [34], la mesure phrasal overlap est une méthode qui compare deux phrases en calculant le nombre de bigrammes (paires de mots consécutifs) qu'elles partagent. Cette mesure prend en compte non seulement la présence de mots individuels, mais aussi leur position relative dans la phrase. Plus il y a de bigrammes en commun entre deux phrases, plus leur similarité est élevée. Sa formule est :

$$sim(S1,S2) = \frac{\textbf{nbr bigrammes en commun entre S1 et S2}}{\textbf{nbr total bigrammes dans S1 ou S2}}$$

Où S1 et S2 sont les deux phrases à comparer. Le résultat de cette formule est une valeur comprise entre 0 et 1, où une valeur plus proche de 1 indique une forte similarité entre les phrases.

La mesure de similarité de chevauchement inter-phrase est basée sur l'idée que deux phrases sont similaires si elles partagent de nombreuses phrases communes. Une phrase peut être définie comme une séquence de mots qui transmet un sens qui n'est pas présent lorsque les mots sont considérés individuellement.

Supposons les deux phrases :

Phrase 1 : "Le chat noir est assis sur le tapis" Phrase 2 : "Le chat gris est sur le canapé"

Extraction des groupes de mots (bigrammes) de chaque phrase :

Phrase 1 :{ "Le chat", "chat noir", "noir est", "est assis", "assis sur", "sur le", "le tapis"}

Phrase 2 : {"Le chat", "chat gris", "gris est", "est sur", "sur le", "le canapé"}

Les deux phrases partagent 2 groupes de mots ("Le chat" et "sur le").

sim (phrase 1, phrase 2) =2 / (7 + 6 - 2) = 0,22

Le résultat de 0,22 indique que les deux phrases ont une faible similarité.

3.2. TF-IDF Measures (Mesures TF-IDF)

C'est une méthode courante utilisée pour mesurer la similarité entre deux phrases ou documents en utilisant la représentation vectorielle basée sur la pondération TF-IDF.

En utilisant la pondération TF-IDF, chaque phrase ou document peut être représenté sous forme de vecteur de dimension N, où N est le nombre de termes uniques dans le corpus. La mesure de similarité vectorielle entre deux vecteurs de phrases ou de documents est alors calculée à l'aide d'une métrique de distance vectorielle, telle que la distance euclidienne ou la similarité cosinus.

La formulation selon Allan et al.[35]

$$sim_{tfidf} = \sum \log(tf\,(w,S1) + 1) * \log(tf\,(w,S2) + 1) \log\left(\frac{N+1}{df\,(w)+0.5}\right)$$

Où $tf_{(w,S1)}$ est le nombre de fois où le terme w apparaît dans la phrase 1(resp. $tf_{(w,S2)}$); N est le nombre total de phrases; et df(w) est le nombre de phrase où w apparaît.

Exemple :

Phrase 1 : "Le chat noir est sur le tapis" Phrase 2 : "Le chien brun est sur le canapé"

Voici les vecteurs TF-IDF correspondants pour chaque phrase :

Phrase 1 : [0.301, 0.301, 0.301, 0.602, 0, 0, 0, 0]

Phrase 2 : [0, 0, 0, 0.301, 0.301, 0.301, 0.602, 0.301]

La mesure :

tfidf (Phrase 1, Phrase 2) = (0.301 * 0 + 0.301 * 0 + 0.301 * 0 + 0.602 * 0.301 + 0 * 0.301 + 0 * 0.301 + 0 * 0.602 + 0 * 0.301) / ((0.301^2 + 0.301^2 + 0.301^2 + 0.602^2 + 0^2 + 0^2 + 0^2 + 0^2) * (0^2 + 0^2 + 0^2 + 0.301^2 + 0.301^2 + 0.301^2 + 0.602^2 + 0.301^2))^(1/2)= 0.169

La similarité entre ces deux phrases est de 0.169, ce qui indique qu'elles sont peu similaires.

3.3. Linguistic Measures (Mesures linguistiques)

Ces mesures utilisent des connaissances linguistiques telles que les relations sémantiques entre les mots et leur composition syntaxique pour déterminer la similarité entre les phrases. On retrouve cinq mesures linguistiques : Semantic similarity measure (Mesure de similarité sémantique) ; Word order similarity (Similarité de l'ordre des mots) **;** The Combined Semantic and Syntactic Measures (Mesures sémantiques et syntaxiques combinées) ; Nouns Similarity (Similarité des noms) ; Verbs Similarity (Similarité des verbes.).

1**. semantic similarity (Mesure de similarité sémantique)**

La mesure de similarité sémantique permet d'évaluer la proximité entre des éléments linguistiques en prenant en considération leur signification plutôt que leur forme extérieure. Au fil des années, plusieurs versions de cette mesure ont été élaborées,

En premier lieu, Li et al. (2006) [36] propose une mesure de similarité sémantique pour les phrases appelée Sentence Semantic Similarity (SenSim). Cette mesure se base sur la similarité des concepts sémantiques présents dans les phrases. La méthode SenSim commence par extraire les concepts sémantiques des phrases en utilisant un algorithme de désambiguïsation lexicale, qui permet d'identifier le sens le plus probable de chaque mot dans le contexte de la phrase. Ensuite, les concepts sémantiques extraits sont comparés pour mesurer la similarité entre les phrases. En résumé, la mesure de similarité sémantique des phrases proposée par Li et al. est basée sur la similarité des concepts sémantiques présents dans les phrases, en utilisant une méthode de pondération TF-IDF et la mesure de similarité de cosinus. Elle a été évaluée sur plusieurs ensembles de données de référence et a montré de bons résultats en termes de similarité sémantique des phrases.

La formule est la suivante :

$$\mathbf{SenSim(S1,S2)} = (1-\alpha) * \cos(\boldsymbol{\theta}) + \alpha)$$

Où S1 et S2 représentent deux phrases à comparer ;

Θ est l'angle entre les vecteurs de concepts sémantiques de S1 et S2, calculé en utilisant la mesure de cosinus :

Θ = arccos((Vecteur1 * Vecteur2) / (||Vecteur1|| * ||Vecteur2||))

Où Vecteur1 et Vecteur2 sont les vecteurs sémantiques de chaque phrase, ||Vecteur|| représente la norme du vecteur, et * représente le produit scalaire.

α est un paramètre de pondération qui prend en compte la longueur des phrases S1 et S2, et est défini comme suit :

α = e^(-β * (len(A)+len(B))/2)

Où len(S1) et len(S2) représentent la longueur de S1 et S2, et β est un paramètre de régularisation.

SenSim varie entre 0 et 1, où 0 indique une absence de similarité sémantique et 1 indique une similarité sémantique maximale.

Voici un exemple pour illustrer l'utilisation de la formule SenSim de Li et al. :

Supposons que nous avons deux phrases à comparer :

Phrase 1 : "Le chat dort sur le tapis"

Phrase 2 : "Le chat dort paisiblement sur le tapis"

La représentation vectorielle est:

Le : [0.2, 0.3, -0.1]

chat : [-0.1, 0.4, 0.2]

dort : [0.4, -0.1, 0.5]

sur : [0.3, 0.2, 0.4]

le : [0.2, 0.3, -0.1]

tapis : [0.1, 0.5, 0.1]

paisiblement : [0.5, 0.2, 0.3]

Ensuite, le calcul du vecteur sémantique de chaque phrase en faisant la moyenne des vecteurs de chaque mot dans la phrase. Ainsi, les vecteurs sémantiques des deux phrases sont :

Vecteur sémantique de la phrase 1 : [0.18, 0.27, 0.2]

Vecteur sémantique de la phrase 2 : [0.26, 0.27, 0.27]

Calcule de l'angle θ :

θ = arccos((0.18 * 0.26 + 0.27 * 0.27 + 0.2 * 0.27) / (sqrt(0.18^2 + 0.27^2 + 0.2^2) * sqrt(0.26^2 + 0.27^2 + 0.27^2)))

θ = 0.14 radians (soit environ 8 degrés)

SenSim de Li et al. :

SenSim(Phrase1, Phrase2) = (1 - α) * cos(θ) + α

α = 0,5 pour donner une importance égale à la similarité lexicale et à la similarité sémantique. Alors, nous avons :

SenSim(Phrase1, Phrase2) = (1 - 0.5) * cos(0.14) + 0.5

SenSim(Phrase1, Phrase2) = 0,921

Dans cet exemple, la similarité entre les deux phrases est de 0,921, ce qui indique qu'ils sont très similaire.

En deuxième, Mihalcea et al. (2006) [37] propose une mesure de similarité sémantique pour les phrases appelée Sentence Relatedness (S-R) qui se base sur la similarité de co-occurrence des mots dans les phrases ce qui signifie qu'elle prend en compte à la fois les mots qui sont identiques dans les deux phrases et ceux qui sont similaires en termes de contexte et de co-occurrence.

La formule de la mesure de similarité sémantique S-R est la suivante :

$$\mathbf{SR(S1,S2)} = \frac{(\mathbf{log(f(S1,S2))} + \mathbf{1})}{(\mathbf{log(max(f(S1),f(S2)))} + \mathbf{1})}$$

Où S1 et S2 représentent deux phrases à comparer ; f(S1) représente le nombre total d'occurrences de tous les mots dans la phrase S1 dans un corpus de texte de référence ; f(S2) représente le nombre total d'occurrences de tous les mots dans la phrase S2 dans le même corpus de référence ; f (S1, S2) représente le nombre total d'occurrences des mots communs à la fois dans S1 et S2 dans le même corpus de référence.

Exemple :

Les deux phrases suivantes :

A : "Le chat noir est sur le tapis" B : "Le tapis a un chat noir dessus"

Représentation des phrases sous forme de vecteurs de termes :

Par exemple, si le vocabulaire est ["chat", "noir", "tapis", "sur", "a", "un", "dessus"],

Les vecteurs pour les phrases A et B peuvent être représentés comme suit :

Vecteur(A) = [1, 1, 1, 1, 0, 0, 0]

Vecteur(B) = [1, 1, 1, 0, 1, 1, 1]

f(A,B) = cos(Vecteur(A), Vecteur(B)) = (Vecteur(A) . Vecteur(B)) / (||Vecteur(A)|| * ||Vecteur(B)||) = (1*1* + *1*1 + 1*1* + *1*0 + 0*1* + *0*1 + 0*1) / sqrt((1^2 + 1^2 + 1^2 + 1^2 + 0^2 + 0^2 + 0^2) * (1^2 + 1^2 + 1^2 + 0^2 + 1^2 + 1^2 + 1^2)) = 0.6667

S-R(A,B) = (log(f(A,B)) + 1) / (log(max(f(A),f(B))) + 1) = (log(0.6667) + 1) / (log(max(0.6667, 0.5)) + 1) = 0.8318

Le score de similarité S-R(A,B) est de 0.8318, ce qui indique une similarité sémantique élevée entre les deux phrases.

La version a plus récente a été proposé par Malik et al. (2018) [38] qui se base sur la distance de Levenshtein et la similarité de Jaccard entre les mots des phrases. Cette mesure est appelée Jaccard-Levenshtein (JL) et elle est plus simple que d'autres mesures de similarité sémantique telles que SenSim.

La formule de la mesure de similarité JL est la suivante :

$$\mathbf{JL(S1,S2)} = \frac{(\mathbf{1} - \mathbf{d}(\mathbf{S1,S2})}{\mathbf{max}(|\mathbf{S1}|,|\mathbf{S2}|))} * \mathbf{J(S1,S2)}$$

Où S1 et S2 représentent deux phrases à comparer ; d (S1, S2) représente la distance de Levenshtein entre les deux phrases ; max (|S1|, |S2|) représente la longueur maximale entre les deux phrases S1 et S2 ; J (S1, S2) représente la similarité de Jaccard entre les ensembles de mots de S1 et S2.

Voici un exemple pour illustrer l'utilisation de cette formule :

A = "Le chat mange une souris"

B = "Le chat chasse une souris"

Pour calculer JL (A, B), nous devons d'abord calculer d (A, B) et J (A, B):

d (A, B) = 2 (il faut deux modifications pour transformer "mange" en "chasse")

max (|A|, |B|) = max(4, 4) = 4

J (A, B) = 2 / 5 = 0.4 (il y a 2 mots en commun entre les phrases A et B : "le" et "souris", et 5 mots en tout)

Maintenant, nous pouvons utiliser ces valeurs pour calculer JL (A, B):

JL (A, B) = (1 - 2/4) * 0.4 = 0.2

Donc, la similarité sémantique entre les phrases A et B selon la mesure JL est de 0.2.

Cette formule est plus simple que d'autres formules de mesure de similarité sémantique pour les phrases, mais elle est également moins précise dans certains cas, notamment lorsque les phrases sont de longueur différente ou que les mots sont mal ordonnés.

2. **Word order similarity (Similarité d'ordre des mots)**

Li et al. [39] Ont proposé une mesure de similarité sémantique basée sur l'ordre des mots appelée Word Order Similarity (WOS). Cette mesure prend en compte la similarité de la séquence de mots, mais aussi la similarité de la séquence de parties du discours et la similarité des relations syntaxiques entre les mots.

Le principe de WOS repose sur l'idée que plus les structures syntaxiques de deux phrases sont similaires, plus il est probable qu'elles partagent une signification similaire.

La formule de WOS est la suivante :

$$\mathbf{WOS(S1, S2)} = \frac{(\mathbf{2} * |\mathbf{C}(\mathbf{S1}, \mathbf{S2})|)}{(|\mathbf{S(S1)}| + |\mathbf{S(S2)}|)}$$

Où S1 et S2 sont les deux phrases à comparer ;S(S1) et S(S2) sont les séquences de parties du discours de S1 et S2 ;C(S1,S2) est l'ensemble des paires de mots de S1 et S2 qui ont des relations syntaxiques similaires et qui appartiennent au même groupe nominal ;|C(S1,S2)| représente la taille de l'ensemble C(S1,S2) ; |S(S1)| et |S(S2)| représentent la taille des séquences de parties du discours de S1 et S2.

Voici un exemple de calcul de similarité sémantique basée sur l'ordre des mots (WOS) entre deux phrases :

Phrase 1 : "Le chat noir court sur le mur" Phrase 2 : "Le chat sur le mur court noir"

Phrase 1 : {Le, chat, noir, court, sur, le, mur} Phrase 2 : {Le, chat, sur, le, mur, court, noir}

Nombre de paires d'ordre des mots identiques : 4 (Le, chat), (court, noir), (sur, le), (le, mur)

Taille de l'ensemble 1 : 7

Taille de l'ensemble 2 : 7

Similarité WOS = 4 / (7 + 7) = 0,2857

Dans cet exemple, les deux phrases ont une similarité WOS de 0,2857, ce qui indique une certaine similarité sémantique entre les deux phrases, bien qu'il y ait des différences dans l'ordre des mots.

3. **The Combined Semantic and Syntactic Measures (Les mesures combinées sémantiques et syntaxiques)**

L'approche CSSM combine deux mesures de similarité : la mesure de similarité sémantique basée sur un graphe de connaissances appelée Path Length Measure (PLM) et la mesure de similarité syntaxique basée sur l'ordre des mots appelée Word Order Similarity (WOS) qui est déjà évoquée.

La mesure PLM calcule la distance entre les concepts sémantiques des deux phrases en utilisant un graphe de connaissances tel que WordNet. La distance entre deux concepts est calculée en comptant le nombre de liens nécessaires pour atteindre l'un à partir de l'autre.

La mesure WOS, quant à elle, compare les arbres syntaxiques des deux phrases pour déterminer leur similarité en termes d'ordre des mots.

La première version a été adopté en 2006 par li. Et al. [40] et sa formule est :

$$\mathbf{CSSM(S1, S2) = \alpha * WOS(S1, S2) + \beta * Res(S1, S2) + \gamma * Lin(S1, S2)}$$

Où S1 et S2 sont les deux phrases à comparer.

WOS(S1, S2) est le score de similarité Word Order Similarity entre les deux phrases.

Res(S1, S2) est le score de similarité de Resnik entre les deux phrases.(voir le tableau).

Lin(S1, S2) est le score de similarité de Lin entre les deux phrases.(voir le tableau).

α, β et γ sont des coefficients de pondération qui permettent de régler l'importance relative des différentes mesures de similarité.

Tableau II. 1. Score de similarité 'Resnik' et 'Lin'

La formule	Les informations
Le score de similarité de Resnik **sim_resnik(c1, c2) =** **-log(p(c1, c2))**	c1 et c2 sont les deux concepts à comparer. p(c1, c2) est la probabilité de la plus petite sous-summation commune (LCS) de c1 et c2 dans l'ontologie. C'est la probabilité que deux concepts choisis au hasard dans l'ontologie aient un LCS qui est égal à la LCS de c1 et c2. La valeur -log(p(c1, c2)) est utilisée pour inverser la relation d'ordre de p(c1, c2). Cela signifie que plus la probabilité p(c1, c2) est faible, plus la similarité entre c1 et c2 est grande.
Le score de similarité de Lin **sim_lin(c1, c2) =** **2 * IC(LCS(c1, c2)) / (IC(c1)+IC(c2))**	c1 et c2 sont les deux concepts à comparer. LCS(c1, c2) est la plus petite sous-summation commune (LCS) de c1 et c2 dans l'ontologie. IC(c) est l'information contenue dans le concept c, qui est définie comme -log(p(c)), où p(c) est la probabilité d'occurrence de c dans un corpus de textes. IC(LCS(c1, c2)) est l'information contenue dans la LCS de c1 et c2, qui est définie comme -log(p(LCS(c1, c2))), où p(LCS(c1, c2)) est la probabilité d'occurrence de la LCS de c1 et c2 dans le corpus de textes.

Voici un exemple de calcul de similarité sémantique combinant la mesure sémantique de Resnik et la mesure syntaxique de WOS:

Phrase 1 : "Le chat noir court sur le mur" Phrase 2 : "Le chat sur le mur court noir"

Phrase 1 : {Le, chat, noir, court, sur, le, mur} Phrase 2 : {Le, chat, sur, le, mur, court, noir}

Calcule du score Resnik à partie de wordnet :

Phrase 1 : {Le, chat, noir, court, sur, le, mur} => {the, feline, dark, run, on, the, wall}

Phrase 2 : {Le, chat, sur, le, mur, court, noir} => {the, feline, on, the, wall, run, dark}

Calcul de la similarité syntaxique de WOS La similarité sémantique basée sur l'ordre des mots est calculée comme décrit dans l'exemple précédent.

$$CSSM(P1, P2) = (WOS(P1, P2) + Resnik(P1, P2)) / 2$$

Pour les deux phrases ci-dessus, supposons que la similarité de Resnik soit de 0,8 et la similarité WOS soit de 0,2857. La similarité combinée serait donc :

$$CSSM(P1, P2) = (0,2857 + 0,8) / 2 = 0,5429$$

Dans cet exemple, les deux phrases ont une similarité combinée CSSM de 0,5429, ce qui indique une certaine similarité sémantique et syntaxique entre les deux phrases.

Une deuxième version a été réalisé par Malik et al. (2016) [41] qui est similaire à la mesure CSS de Li et al., mais elle utilise des modèles de réseaux de neurones pour représenter les phrases.

Leur modèle utilise deux réseaux de neurones : un réseau de neurones de codage (encoder) et un réseau de neurones de décodage (décoder). Le réseau de codage est entraîné pour représenter chaque phrase en un vecteur de caractéristiques sémantiques, tandis que le réseau de décodage est entraîné pour produire une phrase à partir de ce vecteur de caractéristiques.

La similarité sémantique entre deux phrases est alors mesurée en comparant la similarité de leurs vecteurs de caractéristiques sémantiques obtenus à partir du réseau de codage, tandis que la similarité syntaxique est mesurée en comparant la similarité entre les phrases reconstruites à partir des vecteurs de caractéristiques sémantiques à l'aide du réseau de décodage.

$$\mathbf{similarity(S1, S2) = \alpha * SemanticSimilarity(S1, S2) + \beta * SyntaxSimilarity(S1, S2)}$$

Où :

S1 et S2 sont les deux phrases à comparer.

SemanticSimilarity(S1, S2) est la mesure de similarité sémantique entre les deux phrases basée sur la similarité de leurs vecteurs de caractéristiques sémantiques obtenus à partir du réseau de codage.

SyntaxSimilarity(S1, S2) est la mesure de similarité syntaxique entre les deux phrases basée sur la similarité entre les phrases reconstruites à partir des vecteurs de caractéristiques sémantiques à l'aide du réseau de décodage.

α et β sont des coefficients de pondération qui permettent de régler l'importance relative des deux mesures de similarité.

Voici un exemple d'utilisation de CSSM de Malik et al.

Phrase 1 :"Le chat mange une souris" Phrase 2 : "Le chien chasse un chat".

Phrase 1: "Le chat", "chat mange", "mange une", "une souris"

Phrase 2: "Le chien", "chien chasse", "chasse un", "un chat"

les valeurs de similarité sémantique et de similarité syntaxique pour chaque paire de n-grammes :

"Le chat" et "Le chien"

{une similarité sémantique de 0.5 ; une similarité syntaxique de 0}.

"chat mange" et "chien chasse"

{Une similarité sémantique de 0 ; une similarité syntaxique de 0}.

"mange une" et "chasse un"

{Une similarité sémantique de 0 ; une similarité syntaxique de 0}.

"une souris" et "un chat"

{Une similarité sémantique de 0 ; une similarité syntaxique de 0}.

Multiplication des valeurs de similarité sémantique et de similarité syntaxique avec les poids correspondants (définis par les auteurs) :

"Le chat" et "Le chien" : 0.5 * 0.7 + 0 * 0.3 = 0.35

"chat mange" et "chien chasse" : 0 * 0.7 + 0 * 0.3 = 0

"mange une" et "chasse un" : 0 * 0.7 + 0 * 0.3 = 0

"une souris" et "un chat" : 0 * 0.7 + 0 * 0.3 = 0

La moyenne pondérée de toutes les paires de n-grammes pour obtenir le score de similarité de la phrase :

(Score de similarité de la phrase 1 + Score de similarité de la phrase 2) / nombre total de n-grammes = (0.35 + 0) / 8 = 0.04375

Le score de similarité de ces deux phrases est donc de 0.04375, ce qui indique qu'elles sont peu similaires selon cette mesure.

4-**Nouns similarity (similarité des noms) :**

C'est une mesure de similarité sémantique qui vise à quantifier la proximité sémantique entre les noms contenus dans les deux phrases. La similarité est calculée en utilisant la méthode de quantification intrinsèque de contenu informatif décrit par Hadj Taieb et al. [42] couplé à la mesure de Lin [43]. Cette mesure est basée sur la taxonomie WordNet ,en utilisant la relation "est une" où chaque nom Ni est représenté par un ensemble de sommets Syn(Ni) appelés synsets. Chaque synset comprend une signification spécifique du nom Ni. Le calcul de similarité des deux phrases S1 et S2 est définie comme suit :

$$SS_{Nouns}(S_1, S_2) = \frac{\sum_{N_i \in Nouns(S_1)} \max_{N_j \in Nouns(S_2)} SemSim\left(N_i, N_j\right)}{\max(|Nouns(S_1)|, |Nouns(S_2)|)}$$

Où Nouns(Si) fait référence à l'ensemble des noms dans la phrase Si. La similarité sémantique entre deux noms Ni et Nj est calculée comme suit :

$$SemSim\left(N_i, N_j\right) = \max_{(c_1, c_2) \in Syn(N_i) \times Syn(N_j)} SimSem(c_1, c_2)$$

$$SemSim(c_1, c_2) = \frac{2 \times IC\left(LCS(c_1, c_2)\right)}{IC(c_1) + IC(c_2)}$$

Où LCS(c1, c2) renvoie le plus petit concept qui englobe à la fois les concepts c1 et c2 dans la taxonomie WordNet, en utilisant la relation "est une". Quant à la fonction IC(ci), elle calcule le contenu informationnel du concept ci en utilisant une méthode de quantification spécifique de Hadj Taieb et al. [42] qui est utilisée pour mesurer la quantité d'informations contenue dans le concept ci dans le contexte donné.

Exemple :

S1="When you make a journey, you travel from one place to another."
S2="A car is a motor vehicle with room for a small number of passengers."

Tableau II. 2. Exemple similarité des noms.

	journey	place	Max semsim(Ni,Nj)
car	0.0	0.246	0.246
motor	0.0	0.249	0.249
Vehicule	0.0	0.767	0.767
Motor vehicule	0.0	0.114	0.114
room	0.043	0.314	0.314
number	0.219	0.407	0.407
passenger	0.0	0.188	0.188
SS(S1,S2)=(0.246+0.249+0.767+0.114+0.314+0.407+0.188)/7=0.326			

5-Verbs similarity(similarité des verbes)

Elle se concentre sur la quantification de la contribution des verbes dans la finale similarité sémantique entre deux phrases. Le processus de calcul prend également le temps verbal en considération. Cette idée est considérée comme originale car elle n'a pas été employés dans d'anciens ouvrages de littérature. Le processus commence par extraire l'ensemble des verbes utilisant VerbNet. Il se déplace ensuite pour convertir chaque verbe dans sa forme infinitive avec l'analyseur morphologique. Après cela, la similitude sémantique entre un couple de verbes est calculée sur la base de la méthode de quantification IC de Hadj Taïeb et al. [42] avec la mesure de Lin [43]. L'utilisation du temps verbal dans une phrase est intégrée dans le processus informatique comme décrit ci-dessous. L'analyseur de VerbNet est utilisé pour

extraire le temps verbal adéquat telles que « VBD » pour le passé et « VBZ » pour le présent avec le pronom singulier. La mesure entre deux phrases S1 et S2 est calculé comme suit :

$$SS_{Verbs}(S_1, S_2) = \frac{\sum_{V_i \in Verbs(S_1)} \max_{V_j \in Verbs(S_2)} SemSim(V_i, V_j)}{\max(|Verbs(S_1)|, |Verbs(S_2)|)}$$

Où Verbs(Si) est l'ensemble des verbes inclus dans la phrase Si. La similarité sémantique entre deux verbes Vi et Vj est calculée comme suit :

$$SemSim(V_i, V_j) = \begin{cases} \max_{(v_1, v_2) \in Syn(V_i) \times Syn(V_j)} SimSem(v_1, v_2) & if\ Tense(V_i) = Tense(V_j) \\ 0 & else \end{cases}$$

Où Syn(Vi) désigne l'ensemble des synsets représentant le verbe Vi dans le verbal « est un » taxonomie de WordNet, et Tense(Vi) désigne le temps verbal au sein du verbe concerné. phrase. Les verbes sont exploités sous leur forme infinitive. De plus, simsem() est défini comme suit :

$$SemSim(v_1, v_2) = \frac{2 \times IC(LCS(v_1, v_2))}{IC(v_1) + IC(v_2)}$$

Où LCS(v1, v2) est une fonction qui renvoie le plus petit subsumer commun au verbe paire (v1, v2) dans la taxonomie verbale WordNet "est une". IC(vi) renvoie les informations contenu du concept vi en utilisant la méthode de quantification [42].

Exemple :

S1="those involved were allegedly told to never speak of the incident again, according to the letter".

S2="the letter said staff was told to never speak of the incident"

Tableau II. 3. Exemple similarité des verbess.

	told	speak	Max semsim(Vi,Vj)
involved	0.0	0.0	0.0
told	1.0	0.752	1.0
speak	0.752	1.0	1.0
SS(S1,S2)=(0+1+1)/3=0.66			

4-Domaines D'applications :

Les mesures de similarité entre phrases ont de nombreuses applications dans différents domaines, notamment :

Recherche d'information : les mesures de similarité entre phrases sont largement utilisées pour améliorer les performances des moteurs de recherche en retournant des résultats plus pertinents pour les requêtes de recherche des utilisateurs.

Traitement de langage naturel : les mesures de similarité entre phrases sont utiles pour plusieurs tâches de traitement de langage naturel, telles que la détection de plagiat, la classification de textes, la paraphrase et la traduction automatique.

Recommandation de produits : les mesures de similarité entre phrases sont utiles pour recommander des produits similaires à ceux que l'utilisateur a consultés ou achetés.

Analyse de sentiment : les mesures de similarité entre phrases sont utiles pour détecter les similitudes et les différences dans les opinions et les émotions exprimées dans les textes.

Évaluation de la qualité du contenu : les mesures de similarité entre phrases sont utiles pour évaluer la qualité du contenu en comparant différentes versions d'un texte ou en comparant différents textes sur un même sujet.

En résumé, les mesures de similarité entre phrases ont une grande variété d'applications dans différents domaines et sont utiles pour améliorer les performances de nombreux systèmes de traitement de texte et de recherche.

5-Conclusion :

Dans ce chapitre, nous avons exposé notre étude qui inclut les mesures de similarités entre phrases. Nous avons décrit la notion de phrase telle qu'elle est présentée dans la littérature, ainsi que les principales méthodes de similarité les plus connues et leur évaluation. Enfin, nous avons expliqué où trouver ces méthodes.

Chapitre III

Implémentation

Chapitre 3 : Implémentation

1-Introduction

Ce chapitre présente notre système de RI qui se base sur la représentation de phrases pour capturer les significations des documents et requêtes de manière plus précise. Car un mot isole avec sa signification n'apporte pas d'information comme une phrase. Nous avons utilisé des méthodes de traitement automatique du langage naturel pour analyser la structure grammaticale des textes et les transformer en vecteurs de phrases, qui peuvent être utilisés pour effectuer une recherche plus efficace. Notre but est d'améliorer la pertinence des résultats de recherche.

Dans ce chapitre nous allons expliquer en détail, les techniques que nous avons utilisées avec des exemples de requêtes et de résultats pour illustrer les avantages de notre système par rapport aux systèmes de recherche d'information traditionnels.

Enfin, nous discuterons des limites et des défis que nous avons rencontrés et des perspectives pour des recherches futures dans ce domaine.

2-Processus

Le schéma ci-dessous résume notre système de RI et présente ces étapes clés. La première étape est celle d'indexation qui consiste à organiser et à structurer de manière systématique le corpus afin de faciliter la recherche et la récupération ultérieure. La deuxième étape est celle d'appariement qui consiste à donner un score de similitude entre les documents et la requête en utilisant des mesures de similarité entre phrases pour fournir des résultats de recherche plus précis et pertinents.

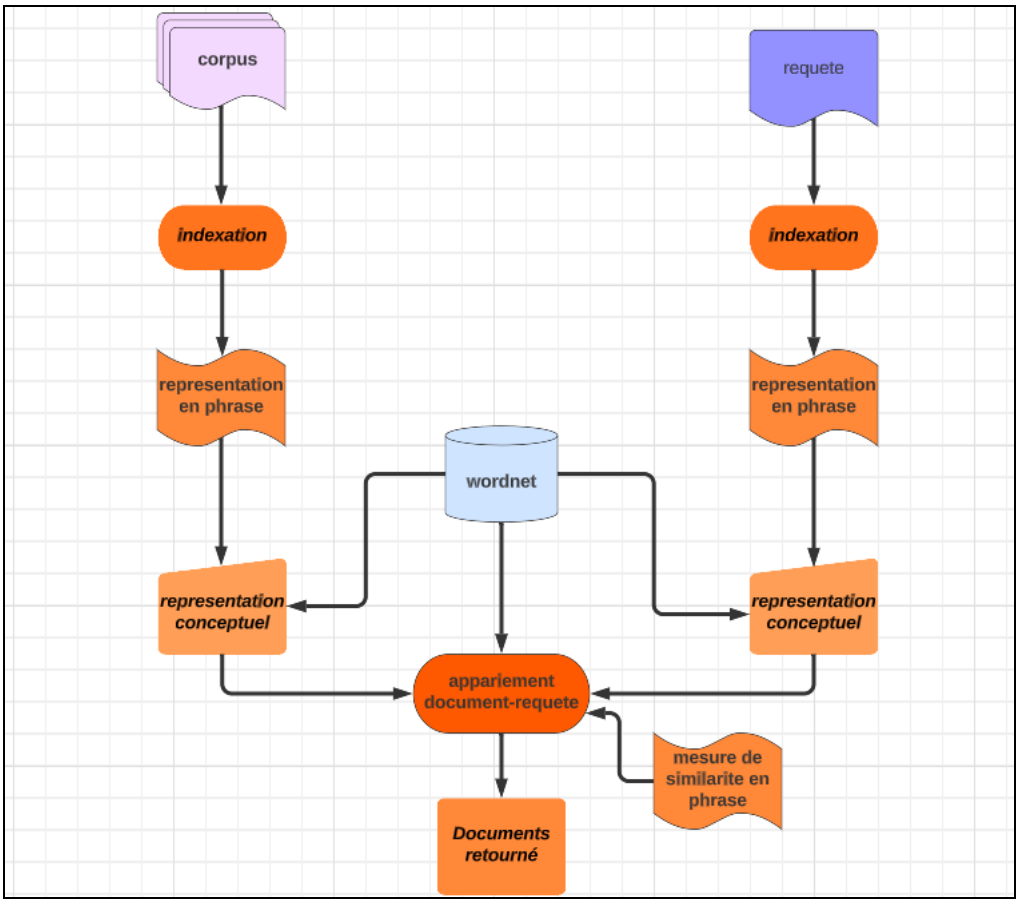


***Figure III. 1** architecture de notre système*

2.1. Indexation :

L'indexation est une étape cruciale dans notre système car elle nous permet de construire une structure de donnée pour représenter le corpus et la requête. Elle suit différentes étapes :

- **Segmentation en phrases :** Cette étape consiste à transformer le texte en phrases distinctes. Nous avons été confrontés à des défis tels que la détermination précise des limites des phrases, notamment en présence d'ambiguïtés ou de constructions spécifiques. Pour résoudre cela, nous avons adopté une approche basée sur des règles syntaxiques visant à identifier les frontières des phrases de manière plus précise.
- **Tokenisation :** Cette étape consiste à transformer chaque phrase en mots ou "tokens". Les tokens sont les unités de base utilisées pour l'analyse linguistique ultérieure. La difficulté de cette étape réside dans la gestion des abréviations, des mots composés et des symboles spéciaux et des entités nommées, qui exigent une analyse contextuelle afin d'effectuer une tokenisation précise.

- **Analyse morphologique :** Cette étape consiste à analyser les caractéristiques morphologiques de chaque token, telles que le genre, le nombre, le temps verbal, etc. Elle permet d'identifier la forme de base (lemme) de chaque mot, ce qui facilite le traitement ultérieur.
- **Étiquetage grammatical :** Cette étape attribue des étiquettes grammaticales à chaque token, indiquant sa partie du discours (par exemple, nom, verbe, adjectif,adverbes).
- **Détection des entités nommées :** Cette étape consiste à identifier et classer les entités nommées. Ils peuvent être des noms de personnes, d'organisations, de lieux, de dates, etc. Cette étape est utile pour l'extraction d'informations spécifiques.
- **Analyse de dépendance :** Cette étape analyse les relations syntaxiques entre les mots de chaque phrase. Elle identifie les dépendances et les liens entre les mots, ce qui permet de comprendre la structure grammaticale de la phrase.
- **Mapping des mots en sens :** Cette étape consiste à associer des significations précises à chaque mot, afin de pouvoir comprendre le sens global de la phrase. Elle implique l'utilisation de ressources lexicales telles que des dictionnaires ou des bases de connaissances pour attribuer des sens spécifiques aux mots. Cependant, de nombreux mots peuvent avoir plusieurs sens possibles. La désambiguïsation est nécessaire pour associer le bon sens à chaque mot en fonction du contexte de la phrase, ce qui contribue à une compréhension plus précise et nuancée du texte.

Pour mieux comprendre ces étapes, considérons le texte suivant :

Texte : "The sun is shining. Birds are singing "in the

1. Segmentation en phrase :

Phrase1 : "The sun is shining."

Phrase2 : "Birds are singing in the trees."

2. Tokenisation sur phrase 1:

Tokens :{"The","sun","is","shining"}

3. Analyse morphologique sur le dernier token :

Token : "shining"

Lemme : "shine"

Caractéristiques morphologiques :

- Partie du discours : Verb (verbe)
- Temps : Present participle (participe présent)

4. Etiquetage grammatical :

- "The" : un déterminant (Dt).
- "sun" : un nom (Noun).
- "is" : un verbe (Verb).
- shining" : un participe présent (Verb).

TableauIII.1. Étiquetage grammatical de JFreeling

Étiquette	**Description**
AO	Adjectif ordinal
AQ	Adjectif qualificatif
CC	Conjonction de coordination
CS	Conjonction de subordination
DA	Déterminant article
DD	Déterminant démonstratif
DI	Déterminant indéfini
DN	Déterminant numéral
DO	Déterminant ordinaux
DP	Déterminant possessif
DT	Déterminant interrogatif/relatif
Fc	Ponctuation de clôture
Fd	Ponctuation de dialogue
Fe	Ponctuation d'ouverture de guillemets
Fg	Ponctuation de fermeture de guillemets
Fh	Ponctuation de tiret
VBG	**pos**:*verb*; **vform**:*gerund*
VB	**pos**:*verb*; **vform**:*infinitive*
VBN	**pos**:*verb*; **vform**:*participle*
VBD	**pos**:*verb*; **vform**:*past*
VBP	**pos**:*verb*; **vform**:*personal*
VBZ	**pos**:*verb*; **vform**:*personal*; **person**:*3*

5. Détection des entités nommées : Dans cette phrase, il n'y a pas d'entités nommées spécifiques.
6. Analyse de dépendance :

   - "sun" est le sujet du verbe "is".
   - "is" est le verbe principal de la phrase.
   - "shining" est un complément du verbe "is".

7. Mapping des mots en sens :

   - "sun" sera associé à la signification du corps céleste qui éclaire la Terre pendant la journée.
   - "shining" sera associé à l'action de briller ou de donner de la lumière.

La figureIII.2. Illustre le résultat de notre système :

« The research finding indicate a strong correlation between exercise and mental health.regular phyical activity have be show to reduce stress level and improve overall well. »

```
--------phrase0---------
the:DT:
research:NN:00636921-n
finding:NNS:00151497-n
indicate:VBP:00923793-v
a:DT:
strong:JJ:02321009-a
correlation:NN:06032246-n
between:IN:
exercise:NN:00624738-n
and:CC:
mental:JJ:01779986-a
health:NN:14447908-n
.:Fp:
--------phrase1---------
regular:JJ:01959294-a
physical:JJ:01778212-a
activity:NN:14006945-n
have:VBZ:00065639-v
be:VBN:02604760-v
show:VBN:00923793-v
to:TO:
reduce:VB:00241038-v
stress:NN:14376188-n
level:NNS:05093890-n
and:CC:
improve:VB:00205885-v
overall:JJ:01582946-a
well-being:NN:14447525-n
.:Fp:
```

FigureIII.2 *Résultat de notre système.*

Construction d'index : la dernière étape de l'indexation permet de construire un index qui permet d'associer chaque document avec ces phrases (sens).

La figure III.3 décrit index de notre système :

corpus :[[[0, 1, 2, 3, 4, 5]], [[6, 7, 8, 9, 10, 11, 12, 13], [14, 15, 16, 17, 4, 8, 18, 19, 20, 21, 22, 23]], [[24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35], [36, 37, 38, 39, 40, 41, 42, 43, 44, 45]], [[46, 47, 48, 49, 4, 50, 51, 52, 53, 54], [55, 4, 56, 57, 13, 58, 59, 60, 61, 62, 52, 53, 63]], [[64, 4, 65, 66, 17, 67, 34, 68, 18, 69, 70], [17, 65, 45, 71, 72, 73, 74]], [[75, 4, 76, 77, 78, 79, 80], [75, 81, 4, 82, 83, 84, 85], [75, 4, 76, 77, 78, 79, 80], [75, 81, 4, 82, 83, 84, 85]]]

Figure III.3 L'index de notre système.

Nous avons mis en place l'algorithme suivant qui illustre l'étape d'indexation :

```
Début

 Entrée : corpus C.

 Variable : phrase[] P, mot[] M, sens[] S.
     Pour chaque document d dans C faire
          P=la segmentation en phrases(d)
         Pour chaque p dans P faire
                M=tokenisation(p)
              Pour chaque m dans M faire
                  sens s=Extraire les sens de m

                       S=ajouter le sens s adéquat

               Fin
          Fin
     Fin
Fin
```

2.2. Appariement :

Le processus d'appariement consiste à évaluer la similitude entre une requête et des documents en utilisant différentes mesures de similarité entre les phrases (vu dans le chapitre2).

Nous utilisons deux extensions pour la comparaison dans notre système. La première extension consiste à améliorer le coefficient de Jaccard en prenant en compte les relations sémantiques entre les mots. La seconde extension utilise des mesures sémantiques telles que Wupalmer, Path et Lin.

Initialement, Le coefficient de Jaccard original, tel qu'expliqué dans le chapitre 2, ne tient pas compte des relations sémantiques entre les mots, ce qui peut limiter sa précision lors de l'évaluation de la similarité entre les phrases.

L'exemple suivant présente le problème évoqué :

Phrase 1 : "Cats chase mice." Phrase 2 : "Felines track rodents."

Lorsque le score est calculé avec le coefficient de Jaccard entre ces deux phrases, nous comparons simplement les ensembles de mots présents dans chaque phrase, sans tenir compte des relations sémantiques entre eux. Dans ce cas, la similarité calculée pourrait être relativement faible, car les mots spécifiques "cats" et "mice" dans la Phrase 1 ne correspondent pas exactement à "felines" et "rodents" dans la Phrase 2.

jaccard(phrase1,phrase2)=0

Pour surmonter cette limitation, nous avons étendu le coefficient de Jaccard en incluant des informations sémantiques, telles que les synonymes, les hyperonymes et les hyponymes.

- La synonymie fait référence à la relation entre deux mots qui ont des significations très similaires ou identiques. En d'autres termes, ce sont des mots qui peuvent être utilisés de manière interchangeable dans un contexte donné. Par exemple, "Happy" et "joyful" sont des synonymes, car ils désignent tous deux la même chose.
- L'hyponymie est une relation hiérarchique où un mot est plus spécifique qu'un autre. Par exemple, dans la catégorie des "fruits", "apple" et "orange" sont des hyponymes, car ce sont des types spécifiques de fruits.
- L'hyperonymie, quant à elle, est la relation inverse de l'hyponymie. C'est une relation hiérarchique où un mot est plus général qu'un autre. Par exemple, "animal" est l'hyperonyme de "cat", "dog" et "horse", car il englobe ces différentes espèces animales.

En tenant compte de ces relations sémantiques, la similarité entre les phrases peut être évaluée de manière plus précise. Par conséquent, le coefficient de Jaccard amélioré entre les deux phrases mentionnées ci-dessus, on obtient un score plus pertinent car la similarité sera prise en compte en raison de leur relation ("feline" est un hyperonyme (superordonné) de "cat"), respectivement pour "mice" et "rodents".

Jaccard_relation(phrase1, phrase2)=0,33

La figure III.4 présente l'algorithme qui a été mis en place pour le coefficient de Jaccard amélioré :

```
Fonction jaccard:
DEBUT
Entrée :phrase p1,phrase p2.
Variable : double score, int numcommun.
   sens[] N1=Extraire_sensnom(p1)
   sens[] N2=Extraire_sensnom(p2)

 POUR CHAQUE sens s1 DANS N1 FAIRE
     POUR CHAQUE sens s2 DANS N2 FAIRE
         Si (s1==s2) Faire
                  Numcommun= numcommun+1
         Sinon
Ensemble R1=Extraire hyperonyme et hyponyme(s1)
Ensemble R2=Extraire hyperonyme et hyponyme(s2)

       Si (ont_relation(R1,s2)ou ont_relation(R2,s1))
       numcommun= numcommun+1
                FIN SI
         FIN SI
       FIN
    FIN
score <- numcommun/ (TAILLE(s1) + TAILLE(s2) - numcommun)
retourne score
```

FigureIII.4. Algorithme de Jaccard amélioré.

La deuxième extension repose sur l'utilisation de mesures sémantiques telles que WuPalmer, Path et Lin. Ces mesures exploitent des ressources linguistiques telles que WordNet pour évaluer la similarité sémantique entre les sens des mots.

- La similarité Path se réfère à la séquence de relations hiérarchiques entre les sens des mots, permettant de mesurer la similarité entre différentes acceptions ou significations des mots.

- La similarité Lin quantifie la similarité entre les sens des mots en se basant sur les informations contenues dans WordNet, en comparant les similitudes entre les intersections et les unions des sens des mots présents dans deux phrases ou deux mots.

- La similarité Wupalmer prend en compte la structure hiérarchique de WordNet pour calculer un score de similarité entre les sens correspondants de deux mots ou phrases, en tenant compte des relations hiérarchiques et contextuelles entre les sens.

Lorsque on applique les mesures sémantiques à l'exemple précédent, on obtient un score plus pertinent.

semantique(phrase1, phrase2)=0,70

L'utilisation de ces mesures permet une évaluation plus précise de la proximité sémantique entre les phrases en exploitant les informations sémantiques fournies par WordNet.

La figureIII.5 illustre l'algorithme des mesures sémantique :

```
Fonction semantique:
DEBUT
Entrée :phrase p1,phrase p2.
Variable : double score.
   sens[] N1=Extraire_sensnom(p1)
   sens[] N2=Extraire_sensnom(p2)
 POUR CHAQUE sens s1 DANS N1 FAIRE
     POUR CHAQUE sens s2 DANS N2 FAIRE
        Score=wu-palmer(s1,s2)
        Ou
        Score=lin(s1,s2)
        Ou
        Score=path(s1,s2)
      FIN
  FIN
retourne score
FIN
```

FigureIII.5. Algorithme des mesures sémantique.

Les deux extensions sont utilisées pour évaluer la similarité sémantique entre les phrases des documents et celles de la requête. Elles sont essentielles pour trouver les documents les plus pertinents en prenant en compte à la fois la correspondance exacte des phrases et les relations sémantiques entre elles.

L'algorithme présenté dans la figure III.6 a été mis en place pour l'étape d'appariement, où il utilise deux fonctions pour calculer la similarité. La première fonction, basée sur la similarité de Jaccard amélioré, évalue les relations sémantiques entre les mots. La seconde fonction quantifie la similarité sémantique entre les sens des mots. En combinant ces deux fonctions, l'algorithme permet d'obtenir une mesure de similarité complète et riche, qui prend en compte

à la fois les relations sémantiques entre les mots ou la similarité entre les sens sémantiques. Cela facilite l'appariement précis et pertinent des données, améliorant ainsi les performances de l'algorithme.

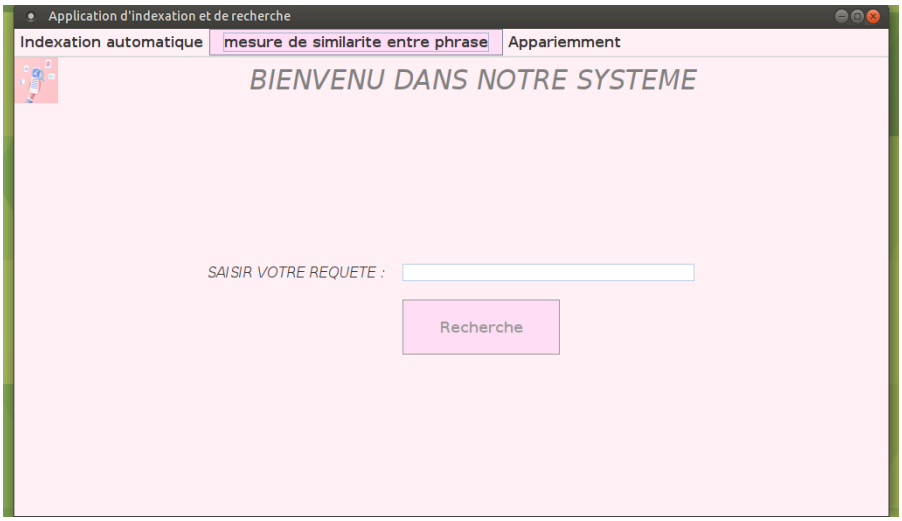
```
Fonction semsim :
DEBUT
Entrée : requête R, index C.
Variable : double[] sim, boolean choix1,choix2.

    POUR CHAQUE document d DANS C FAIRE
        POUR CHAQUE phrase p DANS d FAIRE
           POUR CHAQUE phrase r DANS R FAIRE
            Si (choix1)
            Sim[d]<- sim[d]+jaccard(p, r)
            Si (choix2)
            sim[d]<-sim[d]+semantique(p,r)
        FIN
    FIN
    RETOURNER sim

FIN
```

FigureIII.6. Algorithme d'appariement document-requête.

3-Exemple de déroulement de notre système :

Dans cette section, nous présenterons en détail notre application [APPLICATION DE RECHERCHE]. Cette application a été développée dans le cadre de notre recherche afin de mettre en pratique les concepts et les résultats que nous avons abordés précédemment.

L'interface principale de notre application illustrée dans la figureIII.7 présente une conception conviviale et intuitive pour faciliter l'utilisation. Elle est dotée de trois boutons. Le premier bouton, intitulé "Indexation Automatique », Le deuxième bouton, nommé "Mesure de Similarité entre Phrases", et le dernier intitule "Appariement". L'interface principale dispose également d'un champ de saisie de requête où l'utilisateur peut entrer la requête de recherche. Ce champ de saisie est situé de manière visible et invite l'utilisateur à entrer sa requête pour lancer la recherche.



FigureIII.7. Interface principale

Le bouton d'indexation automatique affiche un menu qui offre à l'utilisateur différentes options pour gérer le processus d'indexation des documents présenté dans la figureIII.8. Il propose trois fonctionnalités principales : l'indexation d'un seul document, l'indexation du corpus et le chargement d'une indexation précédemment réalisée.

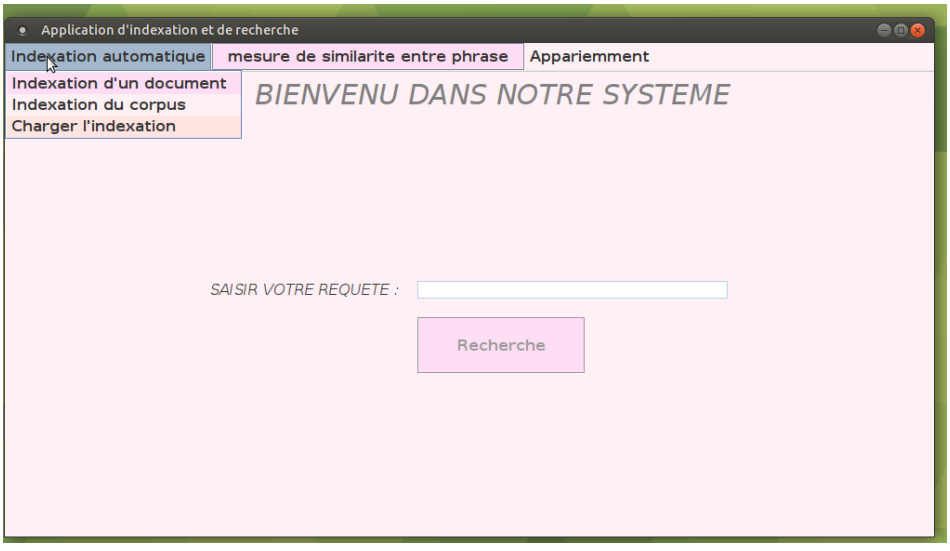
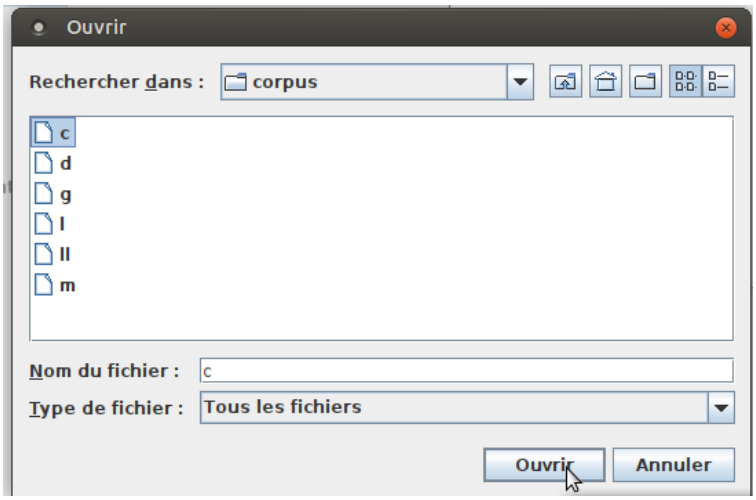
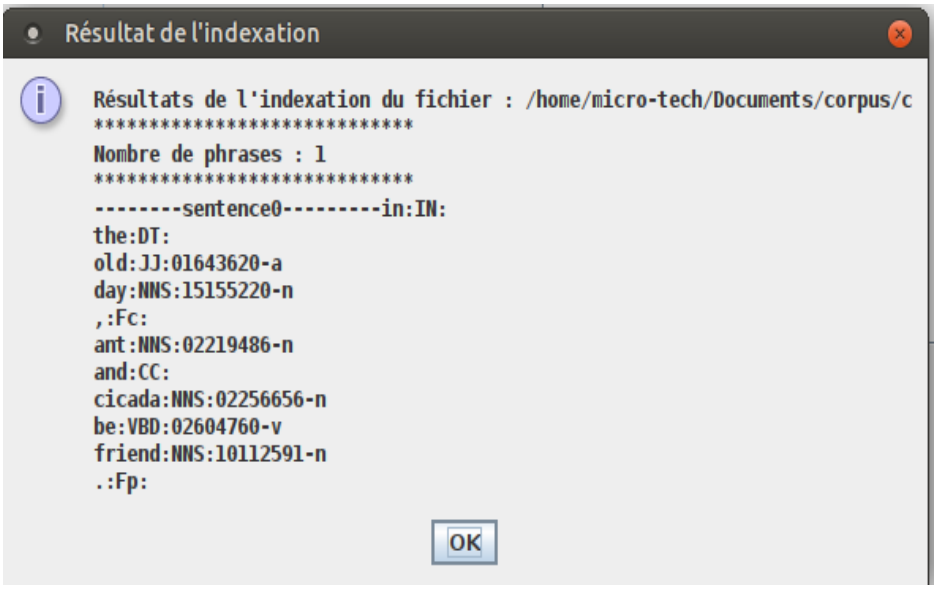


Figure III.8. Menu d'indexation automatique

Lorsque l'utilisateur sélectionne l'option "Indexation d'un seul document", il peut choisir un document spécifique à indexer montrée dans la figureIII.9. Cela lui permet de sélectionner un fichier de lancer le processus d'indexation pour ce document uniquement. Une fois l'indexation terminée, l'application nous retourne le résultat présenté dans la figure III.9.

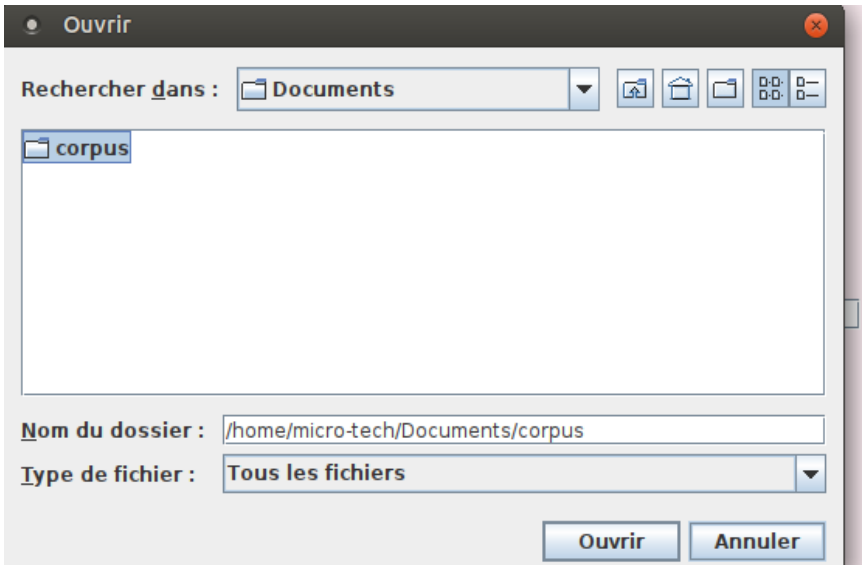


FigureIII.9. Choix du document



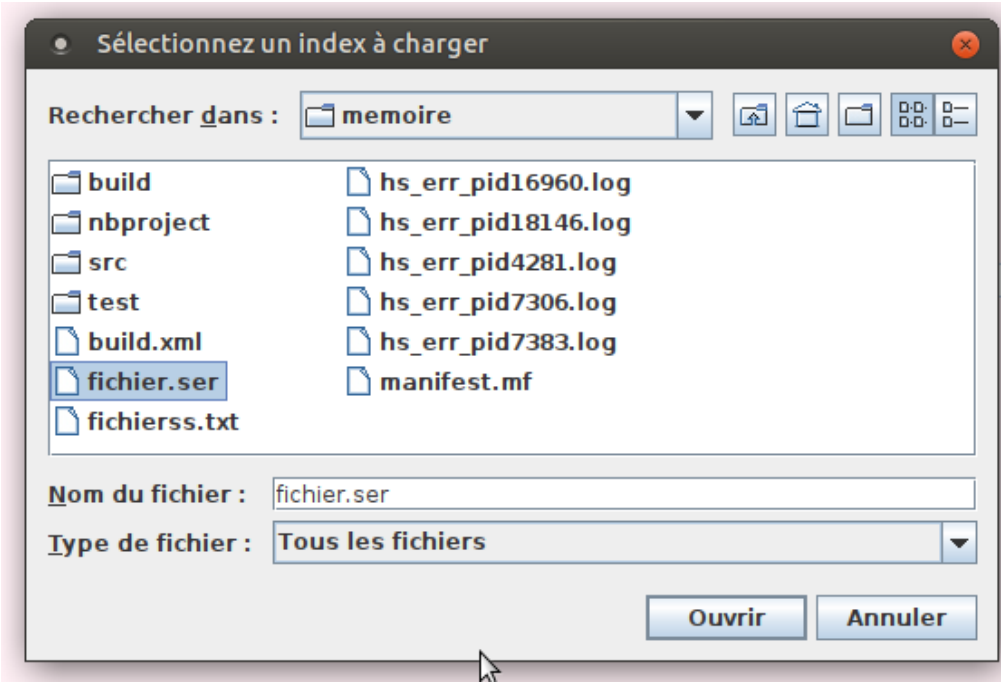
FigureIII.10. Visualisation du résultat de l'indexation.

L'option "Indexation du corpus" permet à l'utilisateur de sélectionner un ensemble de documents à indexer. Il peut choisir un répertoire entier contenant plusieurs fichiers ou spécifier une liste de fichiers à indexer illustrer dans la figureIII.11. Cette fonctionnalité est particulièrement utile lorsque l'utilisateur souhaite indexer un corpus complet. Une fois que l'indexation du corpus est terminée, tous les documents sont prêts à être interrogés simultanément pour des recherches efficaces.



FigureIII.11. Choix du corpus.

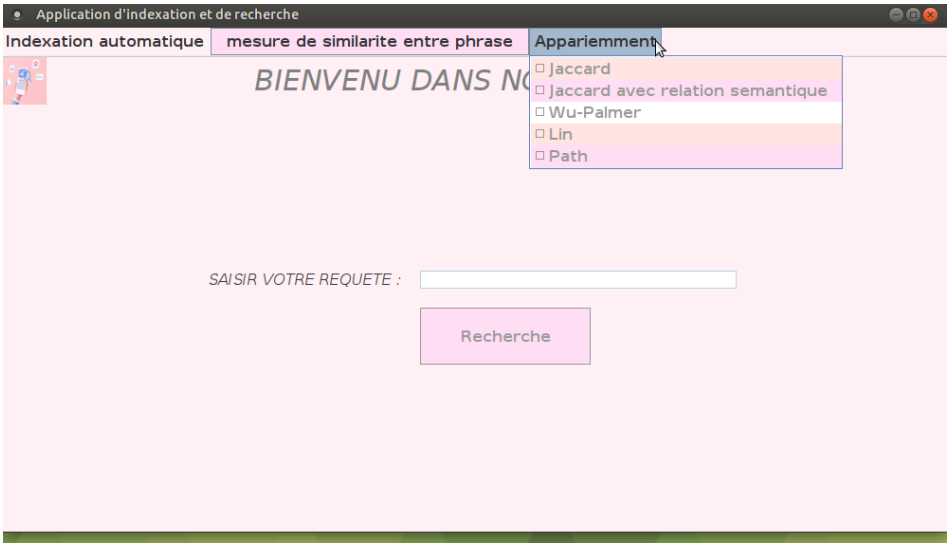
Enfin, l'option "Charger Indexation" permet à l'utilisateur de charger une indexation précédemment réalisée. Si l'utilisateur a déjà effectué une indexation des documents auparavant et souhaite réutiliser cette indexation, il peut utiliser cette option pour charger l'indexation existante comme montrer dans la figureIII.12 et visualiser l'indexation comme présenter dans la figureIII.13. Cela lui évite de devoir recommencer le processus d'indexation à partir de zéro.



FigureIII.12. Fichier d'indexation déjà réalisé.

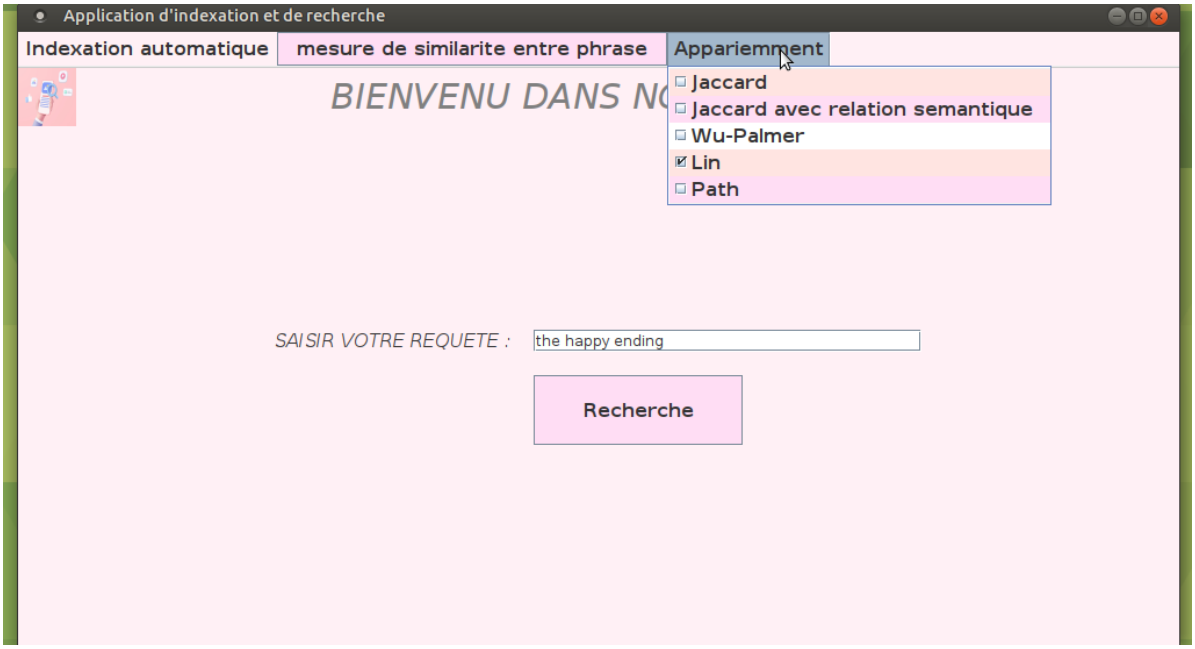
FigureIII.13. Visualisation du résultat.

Le bouton d'appariement de notre interface offre à l'utilisateur plusieurs options pour sélectionner la méthode de similarité à utiliser lors du processus d'appariement entre la requête et les documents indexés illustrer dans la figureIII.14. Les choix disponibles comprennent Jaccard, Jaccard avec relation, Path, Lin et WuPalmer.



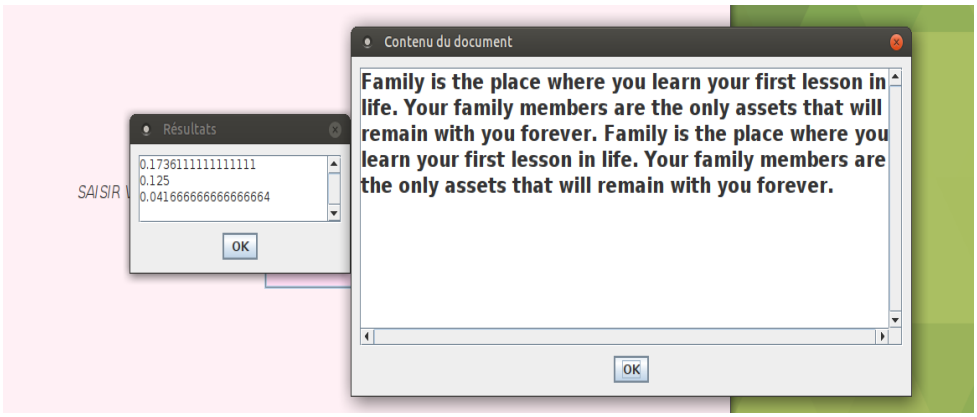
FigureIII.14. Menu d'options d'appariement

Une fois que l'indexation du corpus est terminée ou que l'indexation préexistante est chargée, l'utilisateur est prêt à effectuer des recherches en saisissant une requête pertinente. Il peut entrer sa requête dans le champ dédié à cet effet, en spécifiant la phrase qu'il souhaite rechercher. Après avoir saisi la requête, l'utilisateur peut passer à l'étape suivante, qui consiste à choisir la méthode d'appariement appropriée dans le menu correspondant montrée dans la figureIII.15.



FigureIII.15. Effectuer la recherche.

Une fois la méthode d'appariement choisie, l'utilisateur peut lancer la recherche. L'application utilisera la méthode d'appariement sélectionnée pour évaluer la similarité entre la requête et les documents indexés. La figureIII.16 illustre affichage des résultats de recherche correspondant à la requête, en classant les documents par ordre de pertinence, Pour faciliter l'exploration des résultats, l'utilisateur peut cliquer sur une similarité spécifique dans la liste des résultats. Lorsqu'il clique sur une similarité, le contenu du document correspondant s'affiche dans une fenêtre ou un volet dédié. Cela permet à l'utilisateur de lire, analyser et interagir avec le contenu du document en question, afin d'obtenir une compréhension plus approfondie et précise du contexte.



FigureIII.16. Affichage du résultat.

Le bouton « Mesure de Similarité entre Phrases », offre à l'utilisateur la possibilité de comparer la similarité entre deux phrases. En cliquant sur ce bouton, l'application présente une interface illustre dans la figureIII.17 où l'utilisateur peut saisir deux phrases et choisir parmi les méthodes de similarité disponibles pour évaluer leur degré de similarité sémantique.

FigureIII.17. Interface de similarité entre phrases.

4-Environnement et outils de développement :

Cette section représente les outils de développement utilisé pour mon travail ainsi que l'environnement :

- ***Système d'exploitation :***

Utilisation du système d'exploitation Ubuntu qui est une distribution populaire du système d'exploitation Linux, basée sur Debian.

- ✓ Il offre une interface graphique conviviale et une vaste gamme d'applications préinstallées pour la productivité et le développement.
- ✓ Il est connu pour sa stabilité, sa sécurité et sa facilité d'utilisation et compatible avec de nombreux outils de développement et prend en charge une grande variété de langages de programmation.

- ***Langage de programmation :***

Notre code a été écrit en langage Java qui est un langage de programmation polyvalent et puissant, développé par Sun Microsystems (maintenant Oracle).

- ✓ Il est conçu pour être portable, ce qui signifie que les programmes écrits en Java peuvent être exécutés sur différentes plates-formes sans nécessiter de modifications.
- ✓ Java est orienté objet et dispose d'une vaste bibliothèque standard qui facilite le développement d'applications.
- ✓ Il est utilisé pour créer des applications de bureau, des applications mobiles Android, des applications Web, des services Web, des jeux, etc.

- ***IDE (Environnement de développement intégré) :***

Le logiciel utilisé est NetBeans un IDE open source et multiplateforme pour le développement d'applications Java, ainsi que pour d'autres langages tels que HTML5, PHP, C/C++.

- ✓ Il offre une interface conviviale et de nombreuses fonctionnalités pour améliorer la productivité des développeurs.
- ✓ NetBeans propose des fonctionnalités telles que l'achèvement automatique du code, le débogage, l'analyse statique, la gestion de projets, le support Git, etc.

- ***Base lexicale :***

L'implémentation de notre travail est réalisée avec WordNet version 3.0.

- ✓ C'est une base de données lexicale anglaise qui organise les mots en groupes de synonymes appelés synsets.
- ✓ Chaque synset représente un concept sémantique et contient une liste de mots synonymes.

- ✓ C’est une version spécifique de WordNet, publiée en 2006, qui comprend environ 155 000 synsets et 117 000 mots uniques.Il fournit également des informations sur les relations sémantiques telles que l'hyperonymie (relations hiérarchiques) et l'hyponymie (relations d'inclusion).

- ***Bibliothèque JFReeling :***

JFReeling est une bibliothèque logicielle open source développée par l'Universitat Politècnica de Catalunya (UPC) en Espagne. Elle est conçue pour le traitement automatique du langage naturel (TALN) ce qui signifie qu'il offre des fonctionnalités pour analyser et traiter les données textuelles en langage naturel.

- ✓ JFReeling prend en charge plusieurs langues, notamment l'anglais, l'espagnol, le français, l'italien, le catalan, le portugais et d'autres langues.
- ✓ La bibliothèque fournit des fonctionnalités d'analyse linguistique telles que la segmentation des phrases, la tokenisation, l'étiquetage morphosyntaxique (POS tagging), l'analyse syntaxique (analyse en dépendances), la reconnaissance d'entités nommées, la désambiguïsation lexicale, la lemmatisation, etc.
- ✓ JFReeling utilise des ressources linguistiques telles que des dictionnaires, des modèles statistiques et des règles pour effectuer des analyses linguistiques précises.
- ✓ La bibliothèque est conçue de manière modulaire, ce qui permet aux utilisateurs d'activer ou de désactiver des composants spécifiques en fonction de leurs besoins. Elle offre également des interfaces pour intégrer des modules personnalisés ou des ressources linguistiques supplémentaires.

- ***Bibliothèque JWNL (Java WordNet Library)***

JWNL est une bibliothèque Java qui fournit une interface de programmation pour accéder à WordNet.

- ✓ Elle facilite l'intégration de WordNet dans des applications Java en offrant des classes et des méthodes pour interagir avec les données de WordNet.
- ✓ JWNL permet de rechercher des mots, d'accéder à leurs définitions, de trouver des synonymes, d'explorer les relations sémantiques, etc.

- ✓ Elle simplifie également des opérations plus complexes telles que la recherche de mots ayant des relations spécifiques.

- ***Bibliothèque JWS (Java Wordnet Similarity)***

JWS est une bibliothèque Java spécialement conçue pour calculer la similarité sémantique entre des mots et des concepts de WordNet.

- ✓ Elle fournit plusieurs mesures de similarité couramment utilisées dans le domaine du traitement du langage naturel.
- ✓ Ces mesures de similarité permettent de quantifier le degré de proximité sémantique entre des mots ou des concepts.
- ✓ JWS est utile dans des tâches telles que la recherche d'informations, la traduction automatique, le regroupement de texte, l'exploration de données textuelles, etc.

5-conclusion :

Ce chapitre a présenté notre système de recherche d'information basé sur la représentation en phrase et l'appariement de similarité entre phrases, avec les extensions de coefficient de Jaccard amélioré et de la similarité sémantique entre les sens. Les perspectives de notre système reposent sur l'évaluation approfondie de son efficacité, sur l'optimisation du système en fonction des résultats obtenus, et sur l'incorporation de fonctionnalités supplémentaires pour améliorer l'expérience utilisateur. Ces efforts contribueront à développer un système de recherche d'information performant, adapté aux besoins des utilisateurs et capable de fournir des résultats pertinents.

Conclusion générale

Conclusion générale

Ce mémoire a abordé le défi de l'accès à l'information pertinente et précise dans l'ère de l'explosion des données en développant un système de recherche d'information basé sur l'appariement sémantique. Notre objectif principal était de concevoir et de réaliser un système qui exploite la signification des concepts plutôt que de se limiter aux termes spécifiques utilisés.

Dans cette étude, nous avons adopté une approche méthodologique rigoureuse, en analysant en détail les méthodes d'appariement sémantique existantes et en les adaptant aux besoins spécifiques de notre système. Nous avons présenté les concepts clés pour une compréhension approfondie de la recherche d'information et examiné les différents modèles utilisés, mettant en évidence leurs avantages et leurs défis.

En nous concentrant sur les mesures de similarité entre phrases, nous avons exploré les différentes catégories et présenté les avantages et les inconvénients de chacune. Cette analyse approfondie nous a permis de faire des choix éclairés pour le développement de notre système de recherche d'information.

Notre méthodologie de recherche a été décrite en détail, en expliquant les choix effectués et les raisons qui les sous-tendent. Nous avons présenté les différentes étapes du développement du système, de la collecte des données à la mise en place des techniques d'appariement sémantique. Nous avons également identifié les limites potentielles de notre étude et proposé des stratégies pour les atténuer.

Il convient de souligner que ce mémoire constitue une étape initiale dans le développement d'un système de recherche d'information basé sur l'appariement sémantique. Des travaux futurs sont nécessaires pour évaluer et améliorer davantage notre système. L'évaluation de notre système en utilisant des jeux de données de référence, ainsi que la collecte de commentaires des utilisateurs, nous permettront d'affiner notre approche et de mesurer ses performances.

Conclusion générale

En conclusion, ce mémoire a présenté une approche novatrice pour la recherche d'information en utilisant l'appariement sémantique. L'intégration de la signification des concepts dans notre système ouvre des perspectives passionnantes pour améliorer la pertinence des résultats de recherche. Nous espérons que cette recherche contribuera à l'avancement des systèmes de recherche d'information et à l'amélioration de l'expérience utilisateur dans la récupération de l'information pertinente et précise.

RÉFÉRENCES

[1] SRITI, ALI (2012) *LA RECHERCHE D'INFORMATIONS SEMANTIQUE D'INFORMATIONS DANS LE CADRE DU WEB SEMANTIQUE.* Masters thesis, Université Mohamed Khider Biskra.

[2] Introduction to information retrieval,Christopher D. Manning, Hinrich Schütze, and Prabhakar Raghavan, Publisher, Cambridge University Press, 2008

[3] Modern Information Retrieval, Ricardo Baeza-Yates, Addison Wesley Libri,1999

[4] The dictionary by Merriam-Webster

[5] Reformulation de la requête en recherche d'information en intégrant le profil utilisateur, AOUDJ Leila,masters thesis 2012

[6] Expansion de requête basée sur les phrases, MANSOUR Samia, LAKRIB Sihem,masters thesis 2017 Tizi Ouzou.

[7] https://en.wikipedia.org/wiki/Relevance_(information_retrieval)

[8] Brigitte Simonnot. De la pertinence à l'utilité en recherche d'information : le cas du Web. Viviane Couzinet et Gérard Régimbeau. *Recherches récentes en sciences de l'information : convergences et dynamiques*, ADBS, 2002,

[9] Profils en recherche d'information : définition, exploitation et adaptation ; Anis Ben Ammar. Thèse de doctorat en Informatique,2003

[10] Comparaison des différentes approches de classification automatique des documents textuels ; Sarah DAOUDI et Nassima CHERRAD.Master thesis 2016

[11] Utilisation des Ressources Textuelles Semi-Structurées dans la Recherche Intelligente sur le Web, BOUHADIBA Mohamed el Amine. Master thesis 2015

[12] Un modèle de reformulation des requêtes pour la recherche d'information sur le Web. A. Meftah.2013 Master thesis

[13] Vers Une Méthode D'appariement Document-requête, Multicritères, À Base D'un Réseau De Neurones 2020, AMROUCHE Fatma Zohra et MESSANIA Lamiss.master thesis 2020 Blida

[14] « Modern information retrieval » ; B.-Y. e. Ribeiro-Neto, New York : ACM Press ; Harlow England : Addison-Wesley, cop, 1999.

[15] Implémentation d'un modèle d'appariement pour un Système de Recherche d'Information Personnalisé ; DOUNAS Tarik,OULD FELLA Makhlouf . 2015 master thesis

[16] Extension d'un modèle de recherche d'information pour la prise en compte de la représentation de type wordmbedding ; BOUCHAAL LYDIA, LEULMI Fazia.Master thesis 2017

[17] Institut de Recherche en Informatique de Toulouse https://www.irit.fr/~Mohand.Boughanem/slides/RI/chap4-mod-bool-vect.pdf

[18] extended boolean information retrieval system. salton1983

[19] The SMART retrieval system: Experiments in automatic document processing. Salton G. ; Prentice Hall, 1970.

[20] Recherche d'information sémantique dans les documents XML ; HAMDI Souhila,HENNOUS Souhila.Master thesis 2015

[21] Recherche d'Information : un modèle de langue combinant mots simples et mots composés ; Hammache Arezki. Master thesis 2013

[22] The probability ranking principle in IR. Journal of Documentation, Robertson article 1977

[23] Progress in documentation. Evaluation of information retrieval systems. Journal of Documentation.1970 ;Cleverdon

[24] Mesures de la qualité des systèmes de recherche d'information, Karen Pinel-Sauvagnat, Josiane Mothe article 2013
[25] http://trec.nist.-gov)

[26] http://www.clef-initiati-ve.eu/

[27] http://research.nii.ac.jp/-ntcir/index-en.html

[28] www.wikipedia.org.com

[29] T. McArthur (ed.): The Oxford Companion to the English Language, Oxford University Press, Oxford, 1992.

*[30]*https://vitrinelinguistique.oqlf.gouv.qc.ca/24269/la-grammaire/la-grammaire-actuelle/la-phrase/les-types-de-phrases

*[31]*https://litteratureportesouvertes.wordpress.com/2018/08/22/types-et-formes-de-phrase-quelles-differences/

[32] Jaccard similarity coefficient from https://en.wikipedia.org/wiki/Jaccard_index

[33] Metzler, D., Bernstein, Y., Croft, W., Moffat, A., and Zobel, J. (2005) Similarity measures for tracking information flow. Proceedings of CIKM, 517–524.

[34] Banerjee, S. and Pedersen, T. (2003). Extended gloss overlap as a measure of semantic relatedness. In Proceedings of IJCAI'03, Acapulco, Mexico, 805-810.

[35] Retrieval and Novelty Detection at the Sentence Level, James Allan, Courtney Wade, and Alvaro Bolivar ; Center for Intelligent Information Retrieval ;Department of Computer Science ;University of Massachusetts.

[36] Li, Y., McLean, D., Bandar, Z. A., O'Shea, J. D., and Crockett, K. (2006) Sentence Similarity Based on Semantic Nets and Corpus Statistics. IEEE Transactions on Knowledge and Data Engineering 18, 8, 1138-1150.

[37] Mihalcea, R., Corley, C., and Strapparava, C. (2006) Corpus-based and Knowledge-based Measures of Text Semantic Similarity, in Proceedings of AAAI 2006, Boston, July.

[38] A Combined Semantic and Syntactic Approach for Computing Sentence Similarity Using Deep Learning" publié par Malik et al. en 2018.

[39] Li, S., Abe, N., & Liu, H. (2006). Word order similarity based on syntactic tree kernel for paraphrase identification.

[40] Li, S., Chen, Z., & Li, W. (2006). A combined semantic and syntactic approach to measuring sentence similarity. In Proceedings of the 21st National Conference on Artificial Intelligence (AAAI-06) (pp. 1412-1417). Chicago, IL, USA.

[41] Malik, M. N., Hussain, M., & Lee, S. (2016). A novel semantic and syntactic similarity measure for text classification. Expert Systems with Applications, 60, 1-14.

[42] Hadj Taieb, M. A., Ounelli, H., & Mezghani, N. (2012). An intrinsic content-based term weighting scheme for text classification. In Proceedings of the 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology (Vol. 2, pp. 314-317). IEEE.

[43] Lin, D. (1998). An information-theoretic definition of similarity. In Proceedings of the 15th International Conference on Machine Learning (ICML'98) (pp. 296-304).

[44] The Princeton Encyclopedia of Poetry and Poetics" Roland Greene et al.

[45] Dolan, W., Quirk, C., and Brockett, C. (2004) Unsupervised construction of large paraphrase corpora: Exploiting massively parallel new sources. In Proceedings of the 20th International Conference on Computational Linguistics.

[46] Dagan, I., Glickman, O., and Magnini, B. (2005) The PASCAL recognising textual entailment challenge. In Proceedings of the PASCAL Workshop.

Résume

Ce mémoire présente la réalisation d'un système de recherche d'information basé sur l'appariement sémantique en utilisant les phrases comme choix de descripteurs. L'objectif principal est d'améliorer les résultats de recherche grâce à la sémantique. En effet, le système repose sur deux extensions clés. Tout d'abord, une amélioration de la mesure de similarité de Jaccard a été introduite en ajoutant des relations sémantiques entre les termes des phrases. Ensuite, une deuxième extension a été mise en place, consistant à utiliser des mesures sémantiques entre les sens des phrases.

Mots-clés : recherche d'information, hyperonymes, hyponymes, Jaccard, mesure sémantique.

Abstract

This thesis presents the development of an information retrieval system based on semantic matching using sentences as descriptor choices. The main objective is to enhance search results through semantics. The system relies on two key extensions. Firstly, an enhancement of the Jaccard similarity measure has been introduced by incorporating semantic relationships between the terms in the sentences. Secondly, a second extension has been implemented, which involves using semantic measures between the meanings of the sentences.

Keywords: information retrieval, hypernyms, hyponyms, Jaccard, semantic measure.

ملخص

تتناول هذه الأطروحة تطوير نظام لاسترجاع المعلومات يعتمد على التطابق الدلالي باستخدام الجمل كاختيارات وصفية. يهدف النظام الرئيسي إلى تحسين نتائج البحث من خلال الدلالة. ويعتمد ذلك على امتدادين رئيسيين. أولاً، تم إدخال تحسين على مقياس التشابه جاكارد من خلال إضافة العلاقات الدلالية بين مصطلحات الجمل. ثم، تم تنفيذ امتداد ثانٍ يتضمن استخدام قياسات دلالية بين معاني الجمل

الكلمات الرئيسية: استرجاع المعلومات، الأكثرية العامة، الأكثرية الخاصة، جاكارد، قياس دلالي