REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mohamed Khider – BISKRA
Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie
Département d'informatique

THÈSE

Présentée pour l'obtention du diplôme de
DOCTORAT EN SCIENCES EN INFORMATIQUE
Option : Informatique

THÈME

Contribution à l'amélioration de la recherche d'information par utilisation des méthodes sémantiques: application à la langue arabe

Présentée par :
Mr. Ahmed Cherif MAZARI

Soutenue
Devant le jury composé de :

M[ed] Chaouki BABAHENINI	Professeur	Université de Biskra	Président
Abdelhamid DJEFFAL	Professeur	Université de Biskra	Rapporteur
Saber BENHARZALLAH	Professeur	Université de Batna2	Examinateur
Rachid SEGHIR	Professeur	Université de Batna2	Examinateur
Fouzi HARRAG	MCA	Université de Sétif	Examinateur
Okba TIBERMACINE	MCA	Université de Biskra	Examinateur

2021/2022

Résumé.

Un système de recherche d'information est un ensemble de programmes et de modules qui sert à interfacer avec l'utilisateur, pour prendre et interpréter une requête, faire la recherche dans l'index et retourner un classement des documents sélectionnés à cet utilisateur. Cependant le plus grand challenge de ce système est qu'il doit faire face au grand volume d'informations multi modales et multilingues disponibles via les bases documentaires ou le web pour trouver celles qui correspondent au mieux aux besoins des utilisateurs.

A travers ce travail, nous avons présenté deux contributions. Dans la première nous avons proposé une nouvelle approche pour la reformulation des requêtes dans le contexte de la recherche d'information en arabe. Le principe est donc de représenter la requête par un arbre sémantique pondéré pour mieux identifier le besoin d'information de l'utilisateur, dont les nœuds représentent les concepts (synsets) reliés par des relations sémantiques. La construction de cet arbre est réalisée par la méthode de la Pseudo-Réinjection de la Pertinence combinée à la ressource sémantique du WordNet Arabe. Les résultats expérimentaux montrent une bonne amélioration dans les performances du système de recherche d'information.

Dans la deuxième contribution, nous avons aussi proposé une nouvelle approche pour la construction d'une collection de test de recherche d'information arabe. L'approche repose sur la combinaison de la méthode de la stratégie de Pooling utilisant les moteurs de recherches et l'algorithme Naïve-Bayes de classification par l'apprentissage automatique. Pour l'expérimentation nous avons créé une nouvelle collection de test composée d'une base documentaire de 632 documents et de 165 requêtes avec leurs jugements de pertinence sous plusieurs topics.

L'expérimentation a également montré l'efficacité du classificateur Bayésien pour la récupération de pertinences des documents, encore plus, il a réalisé des bonnes performances après l'enrichissement sémantique de la base documentaire par le modèle word2vec.

Mots-Clés : Recherche d'information arabe; Reformulation de la requête; Méthodes sémantiques; Collections de test RI arabe; WordNet Arabe; Classificateur Naïve-Bayes; Word2vec

Abstract.

An information retrieval system is a set of programs and modules that is used to interface with the user, to take and interpret a query, search in the index and retrieve a ranking of selected documents to that user. However, the biggest challenge of this system is that it has to cope with the large volume of multimodal and multilingual information available through documentary bases or the web to find those that best match the needs of users.

Through this work, we presented two contributions. In the first, we proposed a new approach of query reformulation in the context of Arabic information retrieval. The principle is therefore to represent the query by a weighted semantic tree to better identify the user's information need, whose nodes represent the concepts (synsets) linked by the semantic relationships. The construction of this tree is carried out by the Pseudo-Relevance Feedback method combined to the semantic resource of the Arabic WordNet. Experimental results show a good improvement in the performance of the information retrieval system.

In the second contribution, we also proposed a new approach for building a test collection of Arabic information retrieval. The approach is based on the combination of the method of Pooling strategy using search engines and the Naive-Bayes algorithm of classification by machine learning. For the experiment, we created a new test collection composed of a documentary base of 632 documents and 165 queries with their relevance judgments under several topics.

The experiment also showed the effectiveness of the Bayesian classifier for retrieving relevance of documents, moreover, it achieved good performances after the semantic enrichment of the documentary base by the word2vec model.

Keywords: Arabic Information Retrieval; Reformulation of the query; Semantic methods; Arabic IR test collections; Arabic WordNet; Naive-Bayes classifier; Word2vec

ملخص.

نظام البحث عن المعلومات هو عبارة عن مجموعة من البرامج و وحدات برمجية ، حيث يتم استعماله للتواصل مع المستخدم، لأخذ وفهم استفسار، البحث في الفهرس، واسترجاع مستندات مرتبة إلى هذا المستخدم. ومع ذلك، فإن التحدي الأكبر لهذا النظام هو أنه يتعين عليه التعامل مع الحجم الكبير من المعلومات متعددة الوسائط ومتعددة اللغات المتاحة من خلال قواعد المستندات أو الويب، للعثور على المستندات التي تتناسب بشكل أفضل مع احتياجات المستخدمين.

من خلال هذا العمل، لقد قدمنا مساهمتين. في الأول ، اقترحنا نهج جديد لإعادة صياغة الاستفسار في سياق البحث عن المعلومات باللغة العربية. اساس المبدأ يكون بتمثيل الاستفسار بواسطة شجرة ذات المعاني أو دلالية مرجحة لتحديد احتياجات المستخدم من المعلومات بشكل أفضل، حيث أن عقد الشجرة تمثل المفاهيم (synsets) المرتبطة بعلاقات المعاني. يتم بناء هذه الشجرة من خلال طريقة شبه الردود ذات الصلة المقترنة بالمورد الدلالي لـ WordNet العربي. تظهر النتائج التجريبية تحسن جيد في أداء نظام البحث عن المعلومات.

في المساهمة الثانية، اقترحنا أيضًا نهجًا جديدًا لبناء مجموعة اختبار للبحث عن المعلومات باللغة العربية. يعتمد النهج على مزيج من طريقة استراتيجية التجميع باستخدام محركات البحث و خوارزمية Naive-Bayes للتصنيف عن طريق التعلم الآلي. بالنسبة للتجربة، أنشأنا مجموعة اختبار جديدة تتكون من قاعدة للمستندات ب 632 مستندًا تحت عدة مواضيع و 165 استفسارًا مع أحكامهم على صلتهم بهاته المستندات.

أظهرت التجربة أيضًا فعالية خوارزمية Naive-Bayes للتصنيف في استرجاع أهمية المستندات. علاوة على ذلك، حققت هاته الخوارزمية أداءً جيدًا بعد الإثراء الدلالي لقاعدة المستندات بنموذج word2vec.

الكلمات المفتاحية : البحث عن المعلومات باللغة العربية؛ إعادة صياغة الاستفسار؛ الطرق الدلالية؛ مجموعة اختبار للبحث عن المعلومات باللغة العربية؛ WordNet العربي؛ المصنف Naive-Bayes؛ Word2vec

Dédicaces

À mes chers parents

À ma chère épouse et mes adorables enfants

À mes frères, mes sœurs et toute ma famille

À tous mes amis

Remerciements

Avant tout, je tiens à remercier notre Dieu tout puissant pour m'avoir donné le courage de réaliser ce modeste travail.

Je tiens vivement à remercier mon directeur de thèse Monsieur Abdelhamid DJEFFAL, Professeur à l'université Mohamed KHIDER, Biskra, pour m'avoir donné la possibilité d'effectuer cette thèse sous sa direction. Je tiens aussi à le remercier pour ses précieux conseils durant mes travaux de thèse, sa patience et la préparation de la soutenance.

J'adresse aussi mes plus sincères remerciements à Monsieur Mohamed Chaouki BABAHENINI, Professeur à l'université Mohamed KHIDER, Biskra, pour m'avoir fait l'honneur de présider le jury de ma soutenance, ainsi qu'aux Messieurs Saber BENHARZALLAH et Rachid SEGHIR, Professeurs à l'université de Batna2, Monsieur Fouzi HARRAG, Maître de conférences à l'université de Sétif et Monsieur Okba TIBERMACINE, Maître de conférences à l'université, Mohamed KHIDER, Biskra, pour avoir accepté d'examiner mon travail et faire partie du jury de ma soutenance.

Je ne manquerais pas à remercier également mes collègues du laboratoire LSEA et du département MI (El Bahi, Karim Rédha, Brahim, Hamza K, Hamza H, Sofiene, Billel, Mohamed M, Abdelmoghni, Ismail, Mohamed O, Mohamed R, SidAhmed, Mounir, Diffalah) ainsi les étudiants du Master ISTW de l'université de Médéa plus spécialement à Ali khoudja M., Hayi Med Y., Khorchef N., BenAllel H. qui m'ont beaucoup aidé dans la collecte, le nettoyage et l'annotation les données textuelles et la préparation des datasets pour les différents tests réalisés durant cette thèse.

Finalement, mes plus vifs remerciements vont aussi à mon très cher ami Ali Gagui pour tous ses encouragements et ses aides.

Table des matières

Chapitre II. Sémantique dans les textes et la recherche d'information sémantique

Chapitre III. Recherche d'information Arabe : Outils et ressources

Chapitre IV. Amélioration de la recherche d'information basée sur l'expansion sémantique des requêtes : application à la langue Arabe

Chapitre V. Création d'une collection de test pour des systèmes de RI arabe basée sur la stratégie de Pooling et l'apprentissage automatique

Liste des figures

Chapitre I. Recherche d'information

Chapitre II. Sémantique dans les textes et la recherche d'information sémantique

Chapitre IV. Amélioration de la recherche d'information basée sur l'expansion sémantique des requêtes : application à la langue Arabe

Chapitre V. Création d'une collection de test pour des systèmes de RI arabe basée sur la stratégie de Pooling et l'apprentissage automatique

Liste des abréviations

ACM	Association for Computing Machinery
AWN	Arabic WordNet (WordNet arabe)
CLEF	Conference and Labs of the Evaluation Forum
FIRE	Forum for Information Retrieval Evaluation
IA	Intelligence Artificielle
IDF	Inverse Document Frequency (fréquence de document inverse)
IR	Information Retrieval (Recherche d'Informations)
MAP	Median Average Precision (Précision moyenne)
OCR	Optical Character Recognition (la reconnaissance optique des caractères)
PFE	Projet de Fin d'étude
POS	Part-Of-Speech (partie du discours)
PRF	Pseudo Relevance Feedback (pseudo-réinjection de la pertinence)
RI	Recherche d'Information
SIGIR	Special Interest Group on Information Retrieval
SRI	Système de Recherche d'Informations
TREC	Text REtrieval Conference
STCT	Short Text Conversation Task (Tâche de conversation en texte court)
SVM	Support Vector Machine (machine à vecteurs de support)
TAL	Traitement Automatique de la Langue
TF	Term Frequency (fréquence du terme)
WSD	Word Sense Disambiguation (Désambiguïsation du sens des mots)

Introduction Générale

Contexte et problématique de recherche

Nous vivons dans l'ère de la révolution informationnelle, où l'information ne cesse de s'accroitre et de s'accumule sous différents supports tels que les documents, les vidéos, les sons, les pages web, les bases de données, etc. Nous arrivons donc à une situation totalement contradictoire; il n'y a jamais eu autant d'informations disponibles, mais chercher et trouver de l'information dans cette masse est devenu plus complexe et laborieux.

Devant cette situation, les moteurs de recherche ou les systèmes de recherche d'information sont devenus des outils informatiques incontournables pour trouver une information quelconque, ce qui a poussé le domaine de la Recherche d'Information (RI) de devenir davantage crucial et très actif.

Le plus grand défi pour ces outils informatiques est donc, de pouvoir trouver les informations qui correspondent au mieux à l'attente de l'utilisateur parmi cette quantité d'information disponible. Ces informations peuvent être éditées sous différents supports numériques tels que les documents textuels et les médias incluant l'image, le son et la vidéo, y compris la parole et la musique, ce qui exige des techniques et des programmes informatiques de devenir plus performants dans la tache de la recherche et de la sélection de l'information, l'indexation et le stockage physique des documents, l'interprétation des requêtes des utilisateurs et la manière d'appariement de ces requêtes avec les index et la rapidité des réponses.

En plus avec l'évolution très rapide et la démocratisation de l'Internet, les Systèmes de Recherche d'Information (SRIs) doivent faire face aussi, après la gestion du volume grandissant de l'information sous la multitude de ses supports numériques, au caractère multilingue de l'information qui représente un défi considérable (Culpepper, Diaz, & Smucker, 2018). Dans ce cadre, l'attention croissante pour les autres langues hors l'anglais

a suscité l'adaptation des SRIs et le développement de nouveaux algorithmes et méthodes, et de nouvelles ressources afin de permettre leur traitement informatique.

Ce besoin est très important face à la proportion des utilisateurs et les internautes se servant de la langue arabe, la langue qui fait partie des cinq les plus parlées au monde, avec plus de 420 millions de locuteurs natifs correspondant au nombre des habitants des 22 pays de la ligue arabe (*le nombre de 300 millions est aussi communiqué par les Nations Unies -2021-*), ce qui estime que l'utilisation de la langue arabe sur le Web est comparable à celles des autres langues internationales comme l'anglais.

Malheureusement dans l'état actuel, la recherche d'information en langue arabe souffre toujours de problèmes majeurs à savoir :

(i) Manque de précisions dans les résultats comparables à ceux des autres langues latines.

(ii) Manque des corpus et des collections de test afin d'examiner des nouvelles techniques et des nouveaux algorithmes adaptés pour la langue arabe, à notre connaissance, hors la collection de TREC-2001[1] multilingue en arabe ou l'une de ses versions payantes (de 2001 jusqu'au 2011) créées par LDC[2] (Linguistic Data Consortium), il n'existe aucune collection gratuite disponible à l'exception des collections spécifiques comme celles de la collection ZAD créée par (Darwish & Oard, 2002), qui est construite à partir du livre Zad Al-Mead[3] et la collection KUNUZ [4] composée des documents des textes arabes extraits du *hadith*[5] *de Sahih al-Bukhari* structurée sous format XML.

(iii) Adaptation aux caractéristiques particulières de la langue arabe qui doit faire face :

- A la richesse flexionnelle et dérivationnelle de la langue arabe ;
- A la nature agglutinante de l'arabe: par laquelle nous trouvons plusieurs affixes collés aux unités lexicales véhiculant plusieurs informations morphosyntaxiques ;
- A la diacritisation (تشكيل الحروف) ou les signes diacritiques pour le texte écrit en arabe, comme certains textes sont diacritisés et d'autres non. En effet, l'absence de ces signes entraîne un nombre important d'ambiguïté morphologique, parfois un seul signe diacritique inverse le sens du mot.

[1] https://catalog.ldc.upenn.edu/LDC2001T55

[2] https://catalog.ldc.upenn.edu/

[3] Zad Al-Mead est un livre écrit par le savant musulman **Ibn al-Qayyim** au sujet de Sira.
"زاد المعاد في هدي خير العباد": كتاب من تأليف ابن قيم الجوزية في خمسة مجلدات، يتناول الفقه وأصوله والسيرة والتاريخ وذكر فيه سيرة الرسول صلى الله عليه وسلم.

[4] http://jarir.tn/kunuzcorpus

[5] **Hadith** : recueil des actes et paroles du prophète Mohammed (sws) et de ses compagnons, à propos de commentaires du saint Coran ou de règles de conduite.

Face à ces défis, les méthodes de recherche d'information nécessitent leur révision et adaptation pour leur utilisation spécifique à la langue. D'une part, le développement des nouvelles approches et des techniques de recherche d'information et d'autre part, l'élaboration et la création des nouvelles collections de test de RI spécifiques à l'arabe.

Plusieurs travaux existants ont apporté des solutions pour résoudre la problématique de la RI arabe, sur les différents niveaux du processus du SRI ; i) soit au niveau de la représentation des documents ou l'indexation (Alnaied, Elbendak, & Bulbul, 2020; Chen & Gey, 2002; Darwish & Ali, 2012; Darwish & Oard, 2002; Dilekh, Benharzallah, & Behloul, 2018; El Mahdaouy, El Alaoui, & Gaussier, 2018; Guirat, Bounhas, & Slimani, 2019; Larkey, Ballesteros, & Connell, 2002); ii) soit au niveau de la représentation de la requête (Abbache, Meziane, Belalem, & Belkredim, 2018; Abderrahim, 2014; Mahgoub, Rashwan, Raafat, Zahran, & Fayek, 2014; Mallat, Zouaghi, Hkiri, & Zrigui, 2013; Shaalan, Al-Sheikh, & Oroumchian, 2012); iii) ou bien au niveau d'appariement entre la requête et le document. Parmi ces travaux, certains se sont particulièrement intéressés à l'amélioration des performances des résultats par l'intégration des méthodes sémantiques au sein des niveaux du processus du SRI.

Pourquoi les méthodes sémantiques ? Les algorithmes dans les SRIs classiques se basaient essentiellement sur la recherche de correspondance entre les mots-clés de la requête et les mots-clés des documents, cependant ce principe s'est vite heurté à la problématique du sens, puisqu'un sens peut être représenté par plusieurs mots et un mot peut signifier plusieurs sens. Par exemple lors de la recherche du mot *animal*, il serait plus intéressant de rechercher aussi les documents contenants le mot *chat*, puisque le *chat* est lié sémantiquement au terme *animal.* Cette problématique du sens a engendré donc le problème de l'ambiguïté qui diminuait les performances des SRIs.

Par ailleurs, les travaux basés sur les méthodes sémantiques utilisent principalement des ressources externes telles que les ontologies ou les corpus sémantiques annotés pour tenter de manipuler les sens qui sont au-delà des séquences de mots ou des unités lexicales. Ces travaux ont proposé des techniques pour l'indexation sémantiques, la reformulation sémantique de la requête ou l'appariement sémantique entre document-requête.

Contributions et objectifs du travail

Dans ce cadre décrit précédemment, notre thèse apporte deux contributions majeures ;

La première contribution consiste a proposé une nouvelle approche pour la recherche d'information sémantique arabe, elle s'intéresse à la reformulation des requêtes à travers des arbres sémantiques. Le principe de cette méthode est de construire un arbre sémantique à partir des mots-clés originaux de la requête initiale en combinant la technique de PRF déjà étudiée par les chercheurs dans la littérature à la ressource sémantique externe du WordNet arabe (AWN Arabic WordNet en anglais). La technique de PRF (Pseudo Relevance feedback, en français Pseudo-Réinjection de la Pertinence) se base sur l'analyse des résultats retournés par le SRI.

Le processus de construction de l'arbre commence par la création des concepts initiaux à partir des mots-clés de la requête originale, puis il ajoute de nouveaux concepts à travers des extensions par des relations sémantiques, telle que la synonymie, l'hyponymie et l'hyperonymie. La ressource du WordNet arabe a été aussi utilisée pour la désambiguïsation des termes, des concepts et pour la création de la hiérarchie de l'arbre sémantique.

Nous avons implémenté cette approche dont les résultats expérimentaux montrent une amélioration dans les performances du SRI autour de 10% dans la précision moyenne MAP (Mean Average Precision en anglais).

La seconde contribution de cette thèse concerne la proposition d'une nouvelle approche pour la création d'une collection de test de RI arabe. Le principe de l'approche se base sur la combinaison de deux méthodes, la méthode de la stratégie de Pooling utilisant les moteurs de recherches et la méthode de classification par Naïve-Bayes de l'apprentissage automatique.

A travers l'implémentation de cette contribution, nous avons créé et mis en ligne une nouvelle collection de test de RI arabe. Cette dernière est composée d'une base documentaire de 632 textes bilingues parallèles (arabe/anglais) ainsi de 165 requêtes avec leurs jugements de pertinence sous plusieurs topics et sous-topics.

Dans cette deuxième contribution, nous avons aussi montré l'efficacité de l'algorithme de Naïve-Bayes pour la récupération des pertinences des documents, après l'application du modèle word2vec pour l'enrichissement sémantique de la base documentaire.

Organisation de la thèse

Cette thèse est organisée en cinq chapitres :

Le premier chapitre introduit l'état de l'art sur la recherche d'information et ses différentes notions de base. Ensuite il décrit l'architecture générale d'un SRI et ses composants qui sont la partie d'indexation de la base documentaire, la partie de la reformulation des requêtes et la partie des modèles d'appariement de document-requête. Puis, le chapitre détaille les techniques et les mesures d'évaluation des SRIs à travers les collections de test. Ensuite il explique la conception et la construction de ces collections de test. Finalement, il présente les différents forums et conférences internationaux dédiés au domaine de la recherche d'information et ses applications.

Le deuxième chapitre est consacré à l'état de l'art sur la sémantique, il commence par la présentation de la définition du sens des mots, les relations sémantiques telles que la synonymie, l'antonymie, hyperonymie, la méronymie, aussi les différentes techniques de la désambiguïsation du sens des mots WSD (Word Sense Disambiguation). Il détaille

également la ressource sémantique du WordNet et ses composants. Ensuite, il expose les récents modèles utilisant la sémantique vectorielle tel que; le Word2vec, le GloVe, le FastText, l'ELMo, le Bert et l'openAI GPT. Plus tard, le chapitre aborde les différents travaux connexes dans la littérature qui traitent la problématique de la recherche d'information sémantique et ses différentes méthodes, d'un côté pour l'indexation sémantique et de l'autre côté pour la reformulation sémantique des requêtes.

Le troisième chapitre s'intéresse à présenter l'importance de la langue arabe, ses caractéristiques, ses défis ainsi que ses différents outils, ressources et softwares qui aident son développement dans le domaine du traitement automatique de la langue (TAL) et la recherche d'information (RI), à savoir les segmenteurs, les lemmatiseurs, les analyseurs morphologiques, les outils de reconnaissance des entités nommées, les ressources comme le WordNet arabe et Wikipedia arabe, ainsi les différents moteurs de recherche qui acceptent le traitement du caractère arabe tels que Lucene, Jirs, Whoosh et Hibernate Search. Enfin le chapitre expose les travaux précédents qui ont étudié la problématique de la RI arabe.

Dans **le quatrième chapitre,** nous présentons notre première contribution relative à la reformulation des requêtes basée sur l'arbre sémantique. Dans la première partie, nous détaillons les différentes étapes ; étape 1 de prétraitement de la requête, étape 2 l'extraction et la pondération des concepts relatifs aux mots-clés de la requête et finalement l'étape 3 qui explique la construction de l'arbre sémantique interprétant le besoin de l'utilisateur. Dans la deuxième partie, nous implémentons l'approche proposée et nous examinons les résultats obtenus après son expérimentation sur une collection de test de RI.

Dans **le cinquième chapitre,** nous exposons notre deuxième contribution consacrée à la création d'une nouvelle collection de test de RI pour la langue arabe. La contribution propose une combinaison de deux méthodes, la stratégie de Pooling et l'algorithme Naïve-Bayes de l'apprentissage automatique. La contribution commence par la construction d'une base documentaire et la création d'une liste des requêtes, ensuite en se basant sur la stratégie de Pooling nous récupérons les jugements de pertinence des documents par rapport à cette liste de requêtes sous plusieurs topics. Puis, nous appliquons l'algorithme Naïve-Bayes afin de tester et d'évaluer ces jugements de pertinence. Enfin, le modèle word2vec est aussi utilisé pour enrichir sémantiquement la collection afin d'améliorer les performances du classificateur bayésien.

Enfin, la **thèse se termine** par une conclusion générale dans laquelle, nous présentons une synthèse des contributions apportées pour la problématique de la RI arabe, ainsi les résultats de leurs applications, de même des perspectives envisageables pour des travaux futurs.

Chapitre I. Recherche d'information

I.1. Introduction

La recherche d'information (RI) est devenue pour la plupart des gens une activité quotidienne, par laquelle un utilisateur accède à un programme de moteur de recherche à travers un ordinateur ou smartphone pour récupérer des informations afin de satisfaire un besoin informationnel. Le scénario habituel implique alors que la personne tape une requête et il reçoit des réponses sous forme d'une liste des documents classés.

Le plus grand défi pour ces programmes est donc de pouvoir, parmi un très grand volume d'informations disponibles, trouver celles qui correspondent au mieux à l'attente de l'utilisateur. Ces informations peuvent être sur des supports de documents textuels (dans une ou plusieurs langues) ou des médias incluant l'image, la vidéos et l'audio, y compris la parole et la musique, ce qui nécessite des programmes adaptés et plus performants. L'ensemble de ces programmes informatiques sont appelés système de recherche d'information (SRI) ou le moteur de recherche. En plus de la tache de recherche et de sélection des informations, la recherche d'information implique la manière d'organisation et de stockage physique des documents, la rapidité des réponses, l'indexation et l'interprétation des besoins des utilisateurs.

Dans ce premier chapitre, nous introduisons d'abord la recherche d'information et ses différents concepts de base. Puis, nous décrivons les composants et l'architecture générale d'un système de recherche d'information basée sur le processus d'indexation, le calcul et la pondération des termes, les modèles d'appariement et de reformulation de requêtes. Ensuite, nous décrivons les techniques et les mesures d'évaluation des systèmes de recherche d'information par l'utilisation des collections de test, nous détaillons ainsi leurs conceptions et constructions. Finalement, Nous présentons les différents forums et conférences dédiés à l'évaluation et au test des nouveaux algorithmes et techniques.

I.2. Recherche d'information

I.2.1. Définition

Gerard Salton, pionnier de la recherche d'information et l'une des figures les plus importante des années 1960 aux années 1990, a proposé la définition suivante en 1968 (Gerald Salton, 1968): « *Information retrieval is a field concerned with the structure, analysis, organization, storage, searching, and retrieval of information*. » La recherche d'information est un domaine qui s'intéresse à la structure, l'analyse, l'organisation, le stockage, la recherche et la récupération d'information.

Dans la littérature, plusieurs d'autres définitions ont été données à savoir :

- La recherche d'information est une branche de l'informatique qui s'intéresse à l'acquisition, l'organisation, le stockage et la sélection d'information.
- La recherche d'information est une activité dont la finalité est de localiser et de délivrer des granules documentaires à un utilisateur en fonction de son besoin en informations.
- La recherche d'information est une discipline de recherche qui intègre des modèles et des techniques dont le but est de faciliter l'accès à l'information pertinente pour un utilisateur ayant un besoin en information.

De ce fait nous pouvons dire que la recherche d'information comprend plusieurs techniques, méthodes et taches sur larges types d'informations et une variété d'applications liées à la recherche.

I.2.2. Un peu d'historique

Depuis les années cinquante des petits systèmes ont été proposés pour des petites collections textuelles à des documents bibliographiques en utilisant les modelés booléens. Puis dans les 1960-1970 des méthodes d'évaluation pour des systèmes de RI ont été développés et des expérimentations sur des collections plus large ont été menés. En 1970 le système de RI expérimental SMART (System for the Mechanical Analysis and Retrieval of Text) a été développé par le groupe dirigé par Gérard Saltan (Gerard Salton, 1971a, 1971b), le système se basait sur l'utilisation du modèle vectoriel, la technique de réinjection de la pertinence (relevance feedback) et la classification de Rocchio (Rocchio, 1971). Dans les années 1980, des techniques de RI ont été évoluées par l'influence de l'apparition de l'Intelligence Artificielle (IA) et les systèmes experts. Dans les années 1990, avec les propositions des nouveaux algorithmes et l'avènement de Internet, la RI a été propulsé en avant-scène avec plusieurs applications. L'arrivé de l'Internet a aussi modifié la problématique de la RI, à cette époque, les chercheurs commençaient à s'intéresser de plus en plus à des documents non textuels et des corpus collectés du web.

Dans les deux dernières décennies (2000-2020) avec l'explosion du volume du web, le domaine de la RI est devenu très actif par l'apparition des moteurs de recherche commerciaux puissant comme Google et Yahoo en intégrant des nouvelles techniques à travers l'analyse des liens des pages web, l'indexation d'informations multimédias (l'image, la vidéo, l'audio et la parole), la Recherche multilingue, les systèmes Question/Réponse, le filtrage d'information, la classification et la catégorisation (clustering) des documents, la fouille des textes, l'extraction d'information,… etc.

I.2.3. Notions de base de la recherche d'information

Le processus général de la RI est schématisé par la Figure I.1 ci-dessous. Dans lequel l'utilisateur exprime son besoin informationnel à travers une requête, puis le programme du système de RI (SRI) met en œuvre un modèle d'appariement ou de comparaison entre cette requête et l'index créé préalablement qui représente la base des documents, ensuite ce

programme du SRI retourne les réponses sous forme d'une liste des documents classés suivant un degré de pertinence.

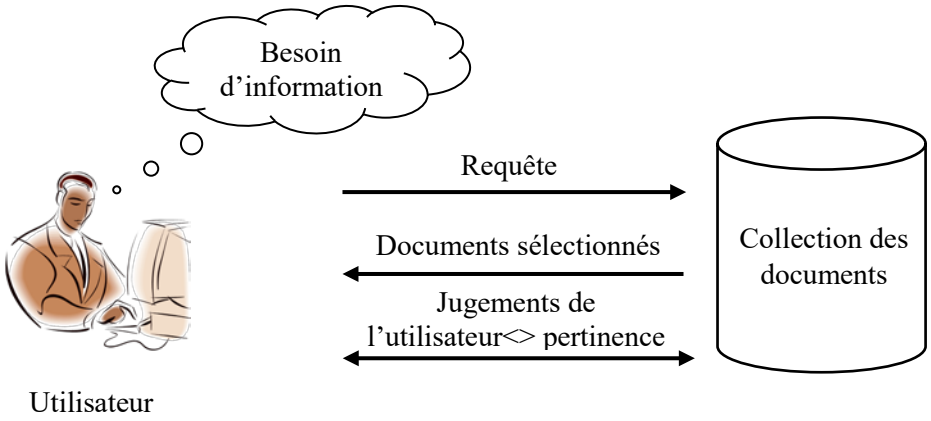

Figure I.1. Processus de RI.

- ***Besoin d'information :*** est une abstraction mentale dont l'utilisateur a besoin pour répondre à une question ou une demande particulière. Ce besoin est exprimé en général sous forme d'un ensemble de mots clés écrits en langage naturel.

Le besoin de l'utilisateur peut être assimilé par plusieurs types de demande comme, chercher une donnée spécifique exemple une date et une heure de vol d'un avion dont le numéro de vol et la compagnie sont connus, chercher un document ou une ressource liée à une thématique ou chercher une réponse inconnue à une question donnée.

- ***Requête :*** est l'interface principale entre l'utilisateur et le programme de recherche. La requête est définie aussi comme la demande formulée par l'utilisateur sous forme d'un ensemble de mots-clés décrits en langage naturel (texte), en langage booléen, en langage graphique à travers l'interface graphique ou même en langage vocal.

- ***Collection documentaire :*** est l'ensemble des documents qui constitue le support des informations ou le corpus. Le document est donc le support élémentaire de l'information qui peut être présenté par un texte, un paragraphe, un livre, une image, une page web ou même une ressource web.

- ***Pertinence et jugement :*** La pertinence d'un document par rapport à une requête donnée est déterminée par un degré de jugement de l'utilisateur. C'est une mesure d'informativité ou un degré de correspondance entre le document à la requête. Dans la littérature la définition de la pertinence en mettant en exergue deux types de pertinence. La pertinence système qui mesure une valeur de similitude entre un document et une requête (Cleverdon, 1970) et la pertinence utilisateur qui est liée à la perception de l'utilisateur sur l'information renvoyée par le système (Harter, 1992; Mizzaro, 1997; Saracevic, 1996). Dans les travaux récents même le profil, la communauté et l'habitude navigationnelle de l'utilisateur sont aussi intégrés dans la pertinence. La pertinence est exprimée par une valeur entre 0 et 1.

I.3. Système de recherche d'information

Un Système de Recherche d'Information (SRI) est un ensemble des programmes qui servent à interfacer avec l'utilisateur, de prendre et interpréter les requêtes, faire la recherche dans l'index, appliquer un modèle d'appariement et retourner les documents sélectionnés à l'utilisateur.

L'architecture des SRIs la plus utilisée dans la littérature est représentée en format U (Belkin, Ingwersen, & Pejtersen, 1992), elle est basée en trois principaux processus (Indexation de la base documentaire, application du model de recherche et analyse de la requête) comme illustre la Figure I.2 suivante :

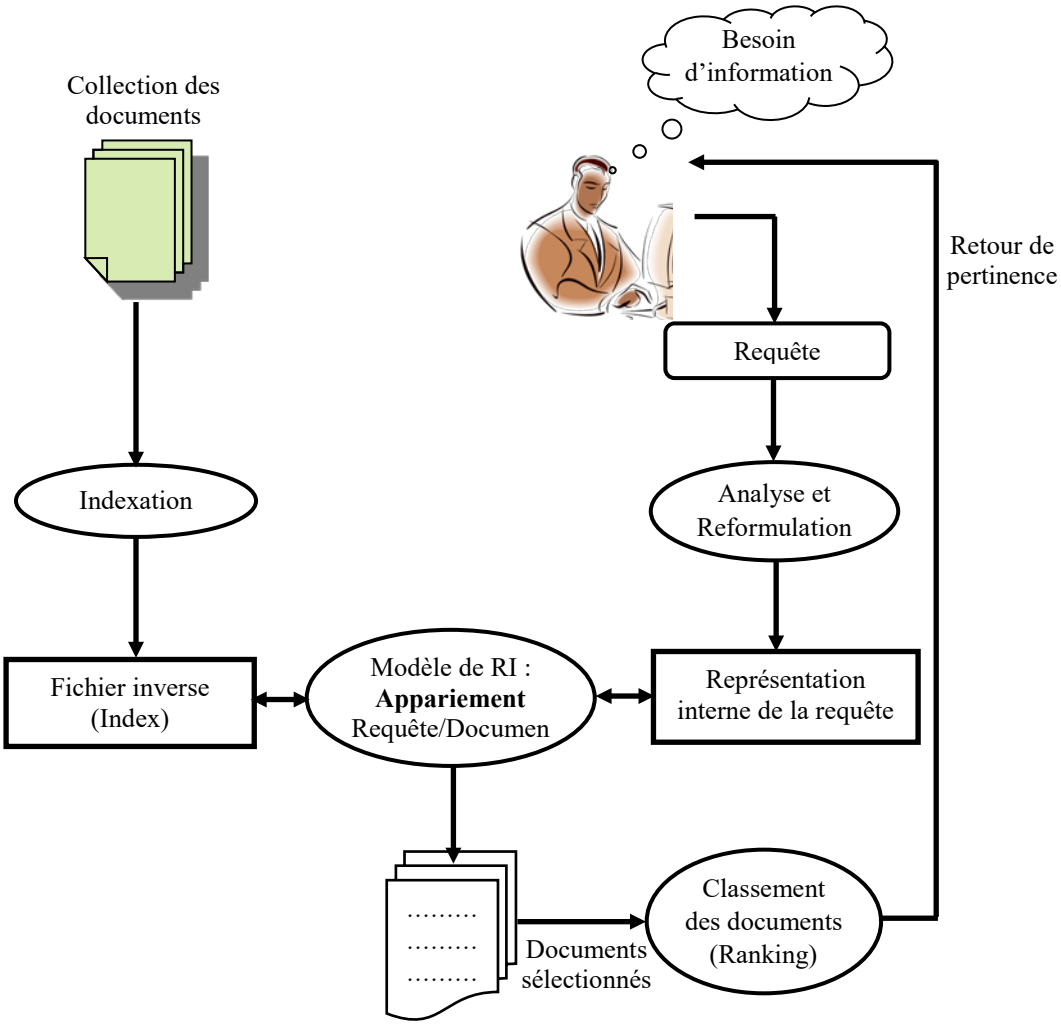

Figure I.2. Architecture générale d'un système de recherche d'information.

Dans ce qui suit, nous décrivons les notions de base qui forment un système de recherche d'information.

I.3.1. Indexation

L'indexation est la procédure de création d'un fichier de descripteurs ou de mots qui représentent le mieux le contenu d'un document, ce fichier est appelé le fichier inverse ou le descripteur. Le processus d'indexation est basé soit sur l'indexation manuelle, dans laquelle des experts choisissent les mots-clés représentatifs du document ou sur l'indexation automatique qui elle-même se base sur les statistiques et le traitement automatique de la langue par un ensemble d'étapes à savoir; la segmentation, la lemmatisation, la suppression des mots vides et la pondération des mots clés (termes) pour enfin créer de l'index.

I.3.2. Pondération des termes

La pondération la plus utilisée dans les systèmes de RI pour représenter les termes et calculer leurs poids dans les document est la pondération TF*IDF et ses dérivés proposée par (Jones, 1972; Gerard Salton & Buckley, 1988). Le principe de cette pondération est de donner un poids plus important et discriminatoire aux termes les plus spécifiques d'un document. Elle repose sur le produit entre la fréquence du terme dans le document (TF : Term Frequency) et la fréquence inverse du document (IDF : Inverse Document Frequency). La fréquence du terme correspond au nombre d'occurrences du terme dans le document et la fréquence inverse du document indique la représentativité globale du terme dans tous les documents de la collection. C'est que veut dire de donner un poids plus important aux termes qui se trouvent moins fréquent dans la collection. La variante de Tf*Idf la plus utilisée est donnée par la formule (I.1).

$$\mathrm{Tf} * \mathrm{Idf} = \begin{cases} (1 + \log_{10}(\mathrm{tf_i})) * \mathrm{Log}_{10}\left(\frac{\mathrm{N}}{\mathrm{n_i}}\right) & \text{si le trme i existe dans le document} \\ 0 & \text{sinon} \end{cases} \quad \text{(I.1)}$$

Où :

N : indique le nombre total des documents dans la collection.

n_i : indique le nombre des documents où apparait le terme i.

Log() : La fonction logarithmique est utilisée pour atténuer les effets de grandes différences entre les fréquences des termes dans le document.

Les fréquences des termes peuvent être aussi normalisées pour réduire les différences entre les valeurs des poids liées à la longueur du document par la formule (I.2) suivante.

$$tf_i = 0.5 + 0.5\ \frac{tf_{ij}}{max(tf_{tj})} \quad \text{(I.2)}$$

Tel que $max(tf_{ij})$ est la plus grande valeur des fréquences des termes dans le document j.

I.3.3. Modèles de RI (Appariement Requête/Document)

Un modèle de RI est un processus d'évaluation et de correspondance en relation avec la représentation de documents et de la requête. Il établit une formalisation du processus de recherche, sélection et classement des documents, son rôle est donc de définir une méthode de comparaison ou d'appariement entre la représentation de document et la représentation de la requête afin de déterminer leur degré de similarité (mesure de pertinence).

Un grand nombre de modèles ont été proposés dans la littérature sous trois principales classes. Entre autre :

Les modèles basés sur la théorie des ensembles : incluant le modèle booléen, le booléen étendu et le modèle basé sur la logique floue.

Les modèles algébriques : fondés sur l'espace vectoriel et le modèle connexionniste.

Les modèles probabilistes : comprenant le modèle probabiliste, le modèle de langages et le modèle de réseau de document ou d'inférence.

Ainsi que d'autres modèles sont aussi proposés comme les modèles basés sur les réseaux d'inférences, les modèles logiques, les modèles neuronaux, les modèles génétiques, d'apprentissage automatique, etc.

I.3.4. Reformulation de la requête

L'analyse et la reformulation de la requête visent à renforcer l'expression de la requête de l'utilisateur (qui est souvent très courte) par l'ajout des nouveaux termes, la suppression de certains termes ou bien la modification des poids des termes dans la requête. De nombreuses approches ont été proposées pour la reformulation des requêtes, parmi les tentatives les plus marquantes, nous retrouvons notamment celles qui exploitent les ressources (comme les thésaurus ou les ontologies), celles qui utilisent le calcul basé sur des co-occurrences ou celles qui se basent sur la technique de la réinjection de la pertinence (relevance feedback), le principe de cette technique est de reformuler la requête par l'ajout des nouveaux termes issus des documents retournés jugés pertinents par les utilisateurs ou bien des documents retrouvés en tête de la liste du résultat de première recherche.

I.4. Évaluation des systèmes de RI

L'objectif du système RI est de localiser et retourner les documents pertinents pour une requête donnée. La qualité d'un SRI peut être donc déterminée en comparant les réponses du système aux réponses idéales attendues par l'utilisateur. Ce qui fait plus les réponses du SRI correspondent aux attentes de l'utilisateur plus le SRI est performant. En conséquence, l'évaluation est déterminée par rapport à la qualité de pertinence des documents retournés par le système de RI.

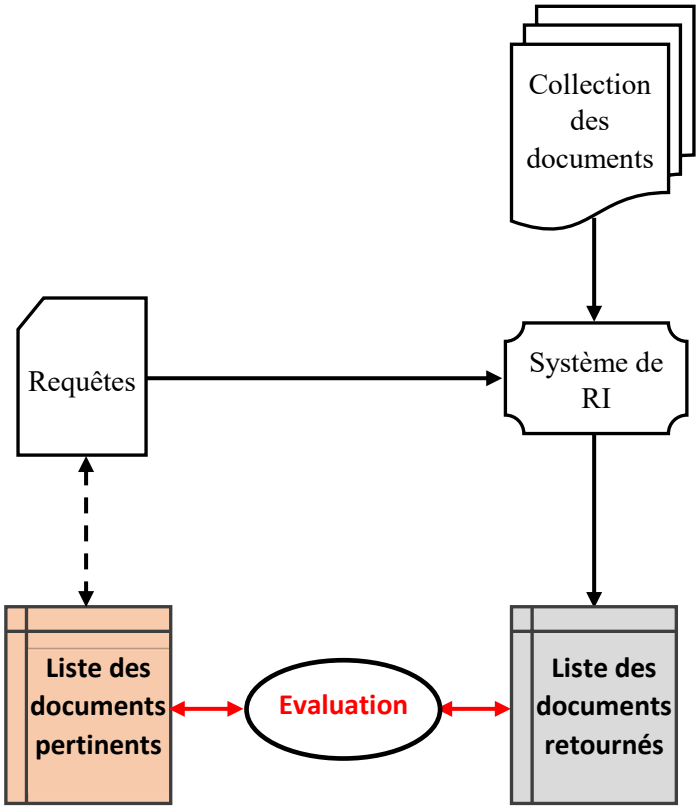

Figure I.3. Evaluation du système de RI.

Pour pouvoir évaluer un système de RI, nous devons alors entrer un ensemble de requêtes, par lequel, le SRI récupère ces requêtes et fait la recherche dans l'index (la collection), ensuite il sélectionne et classe les documents retrouvés dans une liste. L'évaluation n'est donc que le procédé de comparaison de ces résultats des documents récupérés par rapport aux documents réellement jugés préalablement pertinents par les experts. Dans la littérature, plusieurs techniques et métriques ont été proposées pour mesurer les performances et effectuer l'évaluation. L'ensemble des requêtes, le corpus et les jugements de pertinence forment alors la collection de test.

I.4.1. Mesures d'évaluation

Les deux mesures les plus fréquentes et les plus fondamentales pour calculer l'efficacité du système de la recherche d'information sont la précision et le rappel (Figure I.4). Celles-ci sont d'abord définies pour le cas simple où un système RI renvoie un ensemble de documents pour une requête.

I.4.1.1. Précision et rappel

Précision: la précision (P) est calculée par la proportion des documents pertinents sélectionnés par rapport à tous les documents sélectionnés et retournés par le système.

$$Précision = \frac{Nombre\ de\ documents\ pertinents\ sélectionnés}{Nombre\ total\ de\ documents\ sélectionnés} \quad (I.3)$$

Rappel: le rappel (R) est calculé par la proportion des documents pertinents sélectionnés par rapport à tous les documents pertinents dans la collection.

$$Rappel = \frac{Nombre\ de\ documents\ pertinents\ sélectionnés}{Nombre\ total\ de\ documents\ pertinents} \quad \text{(I.4)}$$

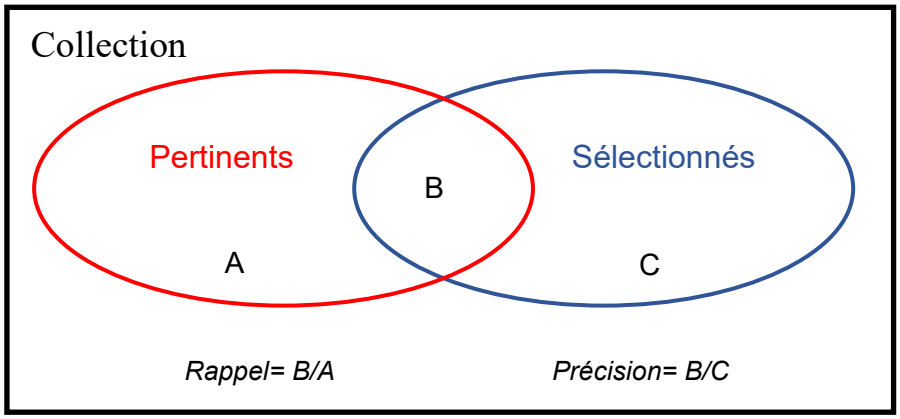

Figure I.4. Précision et rappel.

Pour l'utilisateur, la valeur du rappel indique la capacité du système à trouver des éléments pertinents pour une requête donnée à partir de la collection, la valeur de la précision indique la capacité de trouver les éléments pertinents en top classement selon cette requête.

Idéalement, nous espérons que le système peut fournir une bonne précision et un bon rappel en même temps. Un système idéal avec une précision et un rappel de taux 100% signifie qu'il trouve tous les documents pertinents et rien que les documents pertinents. Cela signifie que les réponses du SRI à chaque requête sont tous des documents pertinents et idéaux par rapport au besoin de l'utilisateur. Ces deux mesures ne sont pas indépendantes. Il existe une relation forte entre elles, quand l'une augmente, l'autre diminue. Cela ne signifie pas d'évaluer la qualité du système en utilisant uniquement une seule mesure. En effet, nous pouvons avoir facilement un taux de rappel de 100% en sélectionnant l'intégralité de la collection comme réponse à chaque requête. Cependant, la précision dans ce cas sera très faible. De même, la précision peut être améliorée en sélectionnant peu de documents pour la réponse, mais le rappel sera diminué. Par conséquent, nous devons utiliser les deux mesures ensemble.

Les métriques de précision et de rappel ne sont pas non plus statiques, (c'est-à-dire qu'un système n'a pas qu'une seule mesure de précision et de rappel), les actions d'un système peuvent évoluer en faveur de la précision ou du rappel (l'une au détriment de l'autre). Par conséquent, plusieurs d'autres mesures sont proposées et utilisées dans la littérature en se basant toujours sur ces deux métriques, à savoir, l'accuracy, la F-mesure, la x-precision, le ROC, ….

I.4.1.2. Métrique d'*Accuracy*

Les notions de Précision et de Rappel peuvent être aussi clarifiées en examinant la matrice de confusion suivante:

	pertinent	**non pertinent**
Sélectionné (Retourné)	true positives (tp)	false positives (fp)
Non-Sélectionné (Non-Retourné)	false negatives (fn)	true negatives (tn)

Où :

$$Précision = \frac{true\ positives}{true\ positives + false\ positives} = \frac{tp}{tp+fp} \quad \text{(I.5)}$$

$$Rappel = \frac{true\ positives}{true\ positives + false\ negatives} = \frac{tp}{tp+fn} \quad \text{(I.6)}$$

Une alternative de métrique qui peut être utilisée pour mesurer les performances d'un système de RI est la métrique d'« Accuracy » (Mot anglais qui veut dire *Exactitude*). Cette mesure n'est que la fraction de ses classifications qui sont correctes. Selon la matrice de confusion ci-dessus, *Accuracy = (tp + tn)/(tp + fp + fn + tn)* (Formule I.7). Cela semble plausible, car il existe deux classes réelles, pertinentes et non pertinentes, et un système de recherche d'information peut être considéré comme un classificateur à deux classes qui tente de les étiqueter comme telle (il récupère le sous-ensemble des documents qu'il juge pertinents). Ce qui fait que la mesure Accuracy est souvent utilisée pour évaluer les systèmes de classification par l'apprentissage automatique.

$$Accuracy = \frac{tp+tn}{tp+fp+fn+tn} \quad \text{(I.7)}$$

Il y a une bonne raison pour laquelle l'Accuracy n'est pas une mesure appropriée pour les problèmes de recherche d'information. Dans presque toutes les circonstances, les catégories des documents sont extrêmement déséquilibrées, plus de 99% des documents sont dans la catégorie non pertinente. Un système avec une forte valeur d'Accuracy peut apparaître performant si en jugeant uniquement tous les documents non pertinents. Même si le système est assez bon, essayer de juger certains documents comme pertinents entraînera presque toujours un taux élevé de faux positifs. Cependant, juger tous les documents comme non pertinents n'est pas du tout satisfaisant pour un utilisateur du système de RI. Les utilisateurs voudront toujours voir certains documents et peuvent être supposés avoir une certaine tolérance pour voir certains faux positifs à condition qu'ils obtiennent des

informations utiles. Les mesures de précision et de rappel concentrent l'évaluation sur le retour des vrais positifs, en demandant quel pourcentage des documents pertinents ont été trouvés et combien de faux positifs ont également été sélectionnés (Manning, Prabhakar, & Hinrich, 2009).

I.4.1.3. F-mesure

La métrique F-mesure est une mesure dérivée de la précision et du rappel. Il s'agit d'une quantité scalaire qui fait un compromis entre la précision et le rappel, qui est la moyenne harmonique pondérée de la précision et du rappel. La formule (I.8) est donnée par (Baeza-Yates & Ribeiro-Neto, 1999; Zhou & Yao, 2010) dans l'équation ci-dessous.

$$F - mesure = \frac{1}{\alpha\frac{1}{p}+(1-\alpha)\frac{1}{R}} = \frac{(\beta^2+1)PR}{\beta^2 P+R} \quad \text{(I.8)}$$

Tel que :

$$\beta^2 = \frac{1-\alpha}{\alpha}$$

$\alpha \in [0, 1]$ et donc $\beta^2 \in [0, +\infty]$. La F-mesure équilibrée par défaut a un poids égal de la précision et le rappel, ce qui signifie que $\alpha = 1/2$ d'où $\beta = 1$. Elle est généralement écrite comme *F1*, qui est l'abréviation de $F_{\beta=1}$, même si la formulation en terme de α de manière plus transparente présente la *F-mesure* en tant que moyenne harmonique pondérée. Lorsque en utilisant $\beta = 1$, la formule se simplifie en (I.9).

$$F - mesure = \frac{2PR}{P+R} \quad \text{(I.9)}$$

Cependant, l'utilisation d'une pondération égale n'est pas le seul choix. Les valeurs de $\beta > 1$ mettent l'accent sur le rappel, tandis que les valeurs de $\beta < 1$ mettent l'accent sur la précision. Le rappel, la précision et la F-mesure sont intrinsèquement des mesures comprises entre 0 et 1, mais elles sont aussi très couramment écrites sous forme de pourcentages, sur une échelle comprise entre 0% et 100%.

I.4.1.4. Courbe de la précision-rappel

La précision, le rappel et la F-mesure sont des mesures basées sur l'ensemble des documents sélectionnés et non ordonnés par le SRI. Il est nécessaire d'étendre ces mesures afin d'évaluer les résultats de recherche classés qui sont maintenant devenus un standard avec les moteurs de recherche. Le principe se repose sur l'analyse des k premiers documents sélectionnés, dans lesquels, pour chaque résultat de recherche, les valeurs de précision et de rappel peuvent être tracées pour donner une courbe appelée la courbe précision-rappel, telle qu'elle est illustrée dans la Figure I.5 suivante.

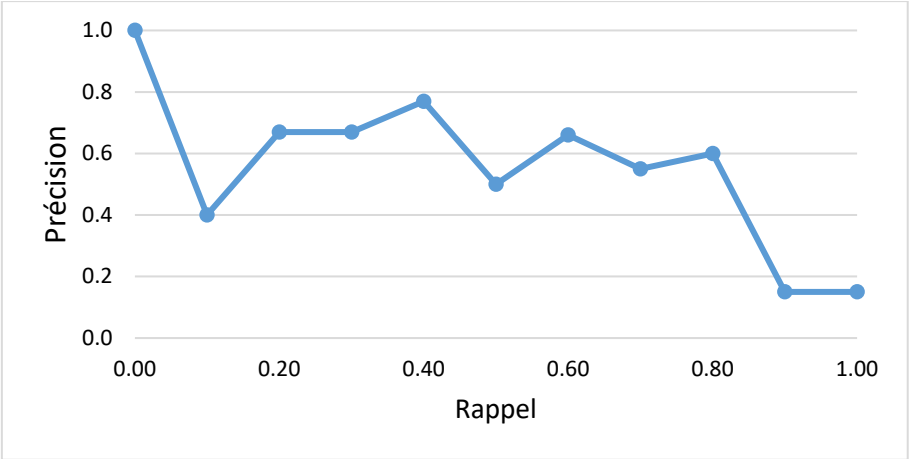

Figure I.5. Courbe de la précision-rappel.

Si le *(k + 1)* ^{ème} document sélectionné n'est pas pertinent, alors le rappel est le même que pour les k premiers documents, mais la précision a chuté ce qui donne une courbe précision-rappel à une forme en dents de scie distinctive. Il est souvent utile de supprimer ces dents par l'application de l'interpolation sur la courbe, cette opération vise à créer une courbe descendante sans forme d'escalier en interpolant la précision pour chaque point des 11 points de rappel *{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0}*. La précision interpolée *($P_{interpolée}$)* est désignée aux différents points de rappel r_i par la formule (I.10), elle est égale à la valeur maximale des précisions obtenues aux points de rappel r, tel que $r >= r_i$.

$$P_{interpolée}(r) = Max_{r \geq r_i} P(r_i) \qquad \text{(I.10)}$$

La Figure 1.6 suivante montre un exemple de l'interpolation de la précision aux 11 points du rappel.

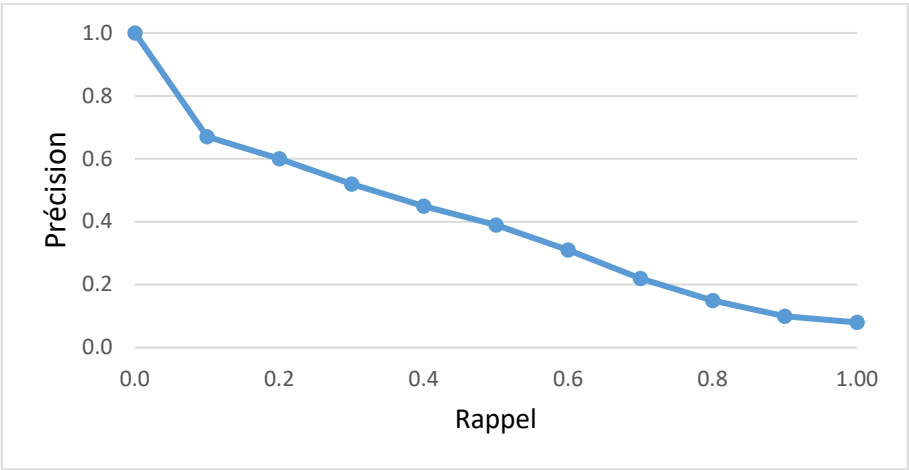

Figure I.6. Courbe de la précision-rappel interpolée.

I.4.1.5. R-précision, x-précision et la précision moyenne

La précision exacte ou la R-précision est obtenue à l'endroit où elle vaut le rappel, par exemple si une requête admet k documents pertinents dans une collection alors la R-précision est égale à la précision au $k^{ème}$ document de la liste des documents sélectionnés et retournés.

Pour la x-précision est la précision au rang x des documents sélectionnés (*x=5, 10, 15, 20, ,* ..etc), notée par *P@x*, est la valeur de la précision calculées à ces différents *x* points.

Une autre métrique est souvent utilisée dans les compagnes d'évaluation qui résume et donne une bonne interprétation des performances des SRI est la précision moyenne (*MAP Median Average Precision)*, par laquelle, elle fournit une mesure de qualité à un seul chiffre pour tous les niveaux de rappel. Parmi les mesures d'évaluation, il a été démontré que la MAP présente une stabilité particulièrement bonne. La valeur de MAP est alors calculée par la moyenne des précisions moyennes non interpolées obtenues sur l'ensemble des documents pertinents sélectionnés pour chaque requête q (Manning et al., 2009). La mesure de MAP est ainsi présentée par la formule (I.11) suivante.

$$MAP(Q) = \frac{1}{|Q|}\sum_{q=1}^{|Q|}\frac{1}{m}\sum_{i=1}^{m_q} Précision\,(R_{qi}) \qquad \text{(I.11)}$$

Où :
Q : est l'ensemble des requêtes.
m : est l'ensemble des documents pertinents pour chaque requête q.
R : document pertinent pour la requête q.

I.4.1.6. Autres mesures

Il existe aussi d'autres mesures d'évaluation des SRIs telles que :

- **Silence et bruit :** ce sont des mesures complémentaires du rappel et de la précision. Le silence est lorsque les documents pertinents ne sont pas sélectionnés par le système, il est égal à (1-Rappel) pour le bruit n'est que les documents sélectionnés et qui ne sont pas pertinents, il est égal à (1- précision).
- **Temps de réponse**: il s'agit du temps nécessaire afin que le SRI sélectionne et classe les documents pour une requête donnée.
- **Satisfaction de l'utilisateur :** il s'agit d'analyser les résultats de pertinence que sur les premiers documents sélectionnés et retournés seulement (10-20 documents), cette mesure est bien utile surtout pour les moteurs de recherche dans le web, où l'utilisateur en général examine que les résultats de la première page de recherche.
- **Présentation des résultats**: certains SRIs peuvent être aussi évalués par la satisfaction de l'utilisateur selon la façon du résultat affiché, il s'agit comment le SRI expose et formule ses résultats de recherche, d'où ces résultats peuvent être visualisés soit sous forme de listes des documents, des liens, des résumés, des pages web,… etc.

I.4.2. Comparaison des systèmes de RI

Afin de comparer deux ou plusieurs systèmes de RI, il est bien utile de les examiner et les évaluer sur les mêmes collections de test (*benchmarks*). Les mesures les plus répondues en pratique (dans la plupart des compagnes d'évaluation à savoir Trec et Clef) pour tester les performances des SRI sont la mesure basée sur des courbes de rappel-précision interpolées et la mesure des moyennes des précisions MAP (*Median Average Precision)*.

Concernant la mesure basée sur des courbes du rappel-précision interpolées, un meilleur système de RI est celui représenté par la courbe qui est supérieure à celle de l'autre système. Si les deux courbes se croisent, il est nécessaire d'analyser les performances suivant les intervalles des points du rappel, il arrive parfois même, de designer difficilement quel est le meilleur système.

A propos de la mesure des moyennes des précisions MAP, l'évaluation est réalisée sur la base de la comparaison directement les valeurs des MAPs, où un taux de performance est déduit par un pourcentage d'amélioration ou de baisse des résultats, comme c'est montré par la formule (I.12).

$$Taux_{performance} = \frac{performance(SRI1) - performance\ (SRI2)}{performance\ (SRI1)}\ X\ 100 \qquad \text{(I.12)}$$

Certaines autres mesures sont aussi utilisées pour évaluer des tests spécifiques mais elles sont moins répondues dans la plupart des compagnes d'évaluation.

I.4.3. Efficacité d'un système de RI versus satisfaction de l'utilisateur

D'autres études dans la littérature ont été menées au-delà de la comparaison des systèmes de RI entre eux, mais d'étudier le lien entre l'efficacité du système de RI par rapport à la satisfaction des utilisateurs pour leurs besoins informationnels, mais ces études ont conclu des résultats différents. Certains parmi ces travaux ont démontré que les résultats obtenus par les SRIs sont fortement corrélés avec la satisfaction des utilisateurs (Al-Maskari, Sanderson, Clough, & Airio, 2008; Sanderson, Paramita, Clough, & Kanoulas, 2010). Par contre, d'autres études ont montré que l'augmentation des performances des SRIs n'ont pas vraiment d'avantage pratique pour les préférences des utilisateurs (Hersh et al., 2000).

Dans les recherches de (Ingwersen & Järvelin, 2006), les auteurs affirment que le véritable problème dans la construction d'un système de recherche d'information n'est pas de savoir si la précision et le rappel augmentent, mais plutôt si le système aide vraiment les utilisateurs à effectuer des tâches de recherche plus efficacement et plus rapide.

En conclusion, le lien entre les mesures de l'efficacité des SRIs et de la satisfaction des besoins d'information des utilisateurs reste un grand axe pour les chercheurs.

I.5. Collection de test des systèmes de RI

Les corpus ou les collections de test se sont fut apparaitre dans les années 70 et qui renferment quelques milliers de documents pour permettre d'évaluer les systèmes de RI qui implémentent de nouveaux modèles et approches, de nouveaux algorithmes, de nouvelles techniques et de nouvelles stratégies de recherche. Les collections de test sont composées principalement d'un ensemble de documents, un ensemble de requêtes et la liste de documents pertinents pour chaque requête (Figure 1.7).

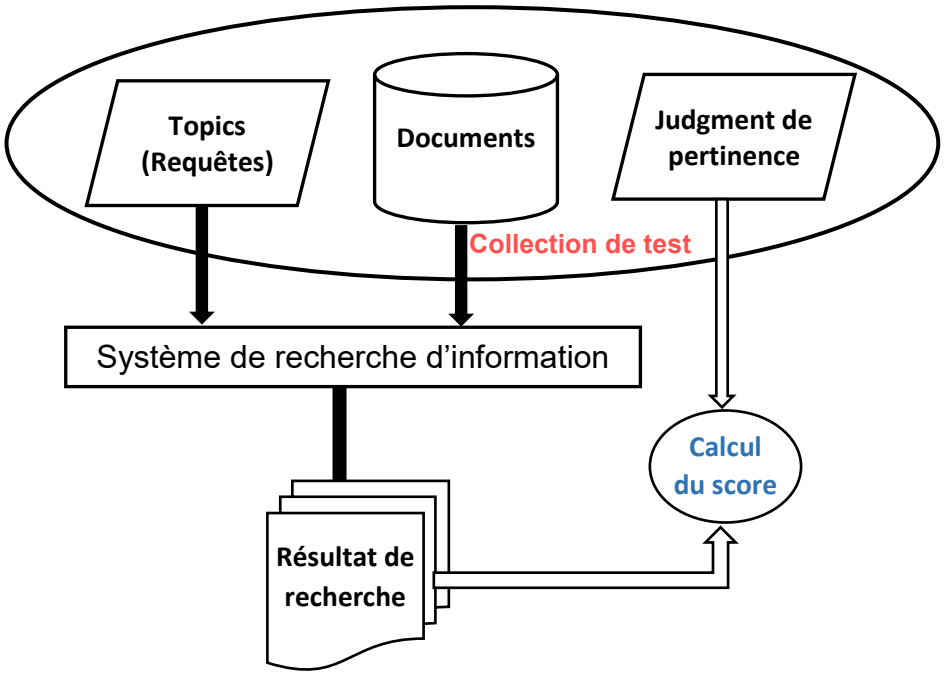

Figure I.7. Collection de test des systèmes de RI.

I.5.1. Conception de la collection de test

Les collections de test sont généralement conçues sur la base du paradigme d'évaluation de *Cranfield* orienté système (Cleverdon, 1997). Ce paradigme consiste à effectuer une évaluation du système de RI dans un environnement de test contrôlé comprenant un corpus de documents, un ensemble de requêtes et des jugements de pertinence. Ainsi, le système classe les documents du plus pertinents au moins pertinents selon les poids du document par rapport à une requête. Ce paradigme de test permet à travers l'expérimentation et d'une manière simple d'évaluer et de tester rapidement les différents nouveaux algorithmes, les nouveaux modèles et les nouvelles méthodes sur le même environnement de test fixe. En plus, ce paradigme permet la comparaison entre les systèmes de RI. Cependant, actuellement certaines applications ne peuvent pas être testées via ce paradigme en raison de la nature des systèmes par leurs contenus dynamiques ou leurs

dépendances aux types d'utilisateurs tels que les systèmes de RI interactifs (liés au profil utilisateur) ou la recherche en temps réel (recherche sur le web).

La conception et le contenu de la collection de test de RI se composent de trois principales parties, un corpus de documents, un ensemble de requêtes et les jugements de pertinence, ces trois parties constituent l'environnement de test, d'où l'évaluation correspond à la capacité de réponse du système de RI de chercher et de récupérer les documents pertinents par rapport aux requêtes.

1. **Corpus de documents** : Les documents peuvent être de différents types tels que du texte (par exemple, des articles de presse, des publications scientifiques, des résumés ou des chapitres de livres, des livres entiers, des textes juridiques ou des pages Web, etc.) ou multimédia (par exemple, des fichiers vocaux, des images ou des vidéos, etc.). Pour la taille de la collection, elle doit de préférence correspondre aux types des systèmes réels à tester, par exemple, le moteur de recherche pour les textes juridiques a besoin des collections formées de quelques milliers de documents, par contre la recherche sur le web a besoin des collections de taille de centaines de millions de documents.
2. **Ensemble de requêtes (par sujet –topic-) :** Les sujets (topics) doivent représenter les besoins d'information des utilisateurs, dans lesquels sont exprimées sous la forme d'un ensemble de requêtes pour chaque sujet (topic).
3. **Jugements de pertinence** : Ce sont les correspondances entre les requêtes et les documents pertinents dans la collection. Le degré et le type de jugement de pertinence peut être aussi décrit et exprimé. La pertinence est évaluée par rapport à un besoin d'information et non à une requête. Par exemple, un besoin d'information peut être: *Informations indiquant si manger du pain gris est plus sain pour réduire le risque du diabète que le pain blanc*. Cela peut être traduit en une requête telle que: « pain ET gris ET blanc ET risque ET diabète ET sain ».

I.5.2. Construction des collections de test

Les corpus de collections doivent contenir des documents représentatifs pour les tâches et les types de systèmes de RI à tester. Cependant, la construction de ces collections de grands corpus est très couteuse suite au coût de jugement de pertinence et au coût de droit d'utilisation et de partage puisque la plupart des corpus récents sont protégés par les droits d'auteurs.

Parfois, des corpus libres et non protégés par le droit d'auteur tels que, Wikipédia et les anciens livres du domaine public sont utilisés, par exemple, comme des ressources disponibles gratuitement pour la constitution de telles collections.

Alternativement, l'acquisition et l'utilisation de corpus peut nécessiter des accords avec les titulaires de droits d'auteur. Par exemple, de nombreuses collections sont construites à partir d'articles de presse récents ou d'articles scientifiques. Les topics ou les sujets des requêtes sont généralement élaborés à partir des besoins d'information réels exprimés par des experts du domaine.

Les collections de test pour les systèmes de RI doivent correspondre alors aux types d'évaluation à réaliser, aux tâches à accomplir et aux algorithmes et techniques à examiner. Ces aspects influent sur les caractéristiques liées aux différentes collections qui sont, les documents de la collection, les jugements de pertinence, les topics ou les sujets, et les requêtes :

I.5.2.1. Documents de la collection

1. **Taille de la collection :** la taille de la collection affecte le comportement et la stratégie de la recherche. Par exemple, dans certaines langues (comme l'arabe ou l'allemand) qui nécessitent une analyse morpholexiale profonde pour déceler l'unité lexicale ou la racine du mot. Dans les premiers travaux, il s'est avéré que la stratégie la plus efficace pour tester des SRIs avaient besoin des petites collections de centaines ou quelques milliers de documents. Cependant, des travaux ultérieurs ont également montré pour l'arabe, que l'utilisation de la lemmatisation légère dans la phase de prétraitement à aider de créer des collections de centaines de milliers ou des millions de documents. En outre, le domaine d'application influe aussi sur la taille des collections, par exemple dans la recherche des documents dans une bibliothèque ou une base documentaire des textes juridiques, la taille de la collection se diffère par rapport à la recherche sur le web où les utilisateurs inspectent généralement que les premiers documents d'une liste classée (Darwish & Magdy, 2014).

2. **Modalité des documents**: les collections peuvent avoir plusieurs différentes modalités selon le type du contenu des documents qui peut être le texte, l'image, la vidéo, l'audio, le graphisme, les données (Rdf, Owl,..), etc.

3. **Type des documents:** le type et la taille des documents de la collection affectent directement les résultats de la recherche. Par exemple, lors de la recherche dans une collection d'articles de presse longue, la normalisation de la longueur des documents est généralement importante. Cependant, lors de la recherche dans une base de commentaires des réseaux sociaux (par exemple des tweets), la normalisation de la longueur du document est moins importante et souvent nuisible. Ainsi, la longueur du document est prise en compte dans la phase de calcul des poids des termes (exemple dans la formule Tf-Idf, la fréquence du terme est en relation avec la longueur du document). Par conséquent, les documents avec des mots rares (avec un faible DF – *Document Frequency:* fréquence des documents contenant le terme) auraient des facteurs de normalisation plus importants (Darwish & Magdy, 2014).

4. **Structure des documents:** la structure de documents peut compliquer ou faciliter la recherche. Par exemple, les articles de presse sont généralement disjoints avec très peu d'éléments structurels (titre, résumé, texte d'article, légendes d'image) qui peuvent être utilisés pour affecter le classement des documents. Les documents Web, quant à eux, présentent de nombreuses caractéristiques structurelles qui peuvent être utilisées pour améliorer le classement (Darwish & Magdy, 2014).

I.5.2.2. Jugement des pertinences

Pour chaque sujet ou topic de la collection de test, un ensemble de jugements de pertinence doit être créé indiquant quels documents de la collection sont pertinents pour chaque topic. La notion de pertinence utilisée dans l'approche de *Cranfield* est généralement interprétée comme une pertinence de topic: « Examiner si un document contient des informations sur le même topic (sujet) que la requête ». De plus, la pertinence est supposée cohérente entre les évaluateurs (des experts qui jugent qu'un document est en relation avec le topic). Cependant, il s'agit d'une vision étroite de la pertinence, qui s'est avérée subjective, situationnelle et multidimensionnelle (Schamber, 1994). Certains ont émis l'hypothèse que cette variabilité permettrait d'évaluer l'exactitude de la mesure de l'efficacité de la recherche et la sélection des documents. Une série d'expériences ont été menées pour tester cette hypothèse (Cleverdon, 1970; Voorhees, 1998) avec des résultats montrant qu'en dépit de différences marquées dans les documents que différents évaluateurs jugeaient pertinents ou non pertinents, les différences n'ont pas sensiblement affecté l'ordre du résultat de sélection relatif aux systèmes de RI mesurés à l'aide des différentes méthodes d'évaluations.

Les jugements de pertinence peuvent être binaires (pertinent ou non pertinent) ou avec des degrés de pertinence, par exemple ; non pertinent, moins pertinent, partiellement pertinent ou très pertinent. L'utilisation des jugements de pertinence gradués par opposition à des jugements de pertinence binaires est importante, car elle a des implications pour lesquelles des mesures d'évaluation peuvent être utilisées pour tester et évaluer les systèmes de RI.

En pratique, il existe deux principales méthodes pour établir des jugements de pertinence, d'où la première méthode est manuelle et la deuxième méthode est semi-automatique.

1. **Méthode manuelle :** La première méthode est manuelle, elle est guidée par l'utilisateur dans laquelle un évaluateur de la pertinence cherche dans le corpus les documents relatifs à un topic (sujet) et puis pour chaque requête il juge manuellement le document (pertinent OU non pertinent). Généralement, les évaluateurs sont des personnes expertes qui ont initialement déjà créé les topics, mais ce n'est pas toujours le cas. Par exemple, certaines évaluations de la pertinence peuvent être aussi recueillies de manière itérative à travers des résultats d'évaluation d'autres évaluateurs. Il faut cependant reconnaître que l'expertise du domaine peut affecter la qualité des évaluations de pertinence obtenues (Bailey et al., 2008; Kinney, Huffman, & Zhai, 2008).
2. **Méthode semi-automatique :** La deuxième méthode est semi-automatique par l'utilisation de la technique de la stratégie de Pooling. Cette technique a été adoptée par TREC dès ses débuts. En fait, l'approche sert à rassembler les N premiers documents résultats des différents moteurs de recherche (ou des systèmes de RI) testés pour chaque topic et à regrouper tous les résultats en une seule liste de résultats à juger (appelée Pooling). Les évaluateurs (expert) de la pertinence parcourent ensuite le pool et émettent des jugements de pertinence sur chaque document. Les documents qui ne sont pas jugés sont souvent classés

comme non pertinents. Un problème avec Pooling est l'exhaustivité des évaluations de la pertinence. Idéalement, pour chaque topic, nous devrons trouver tous les documents pertinents dans la collection de documents; cependant, la technique de Pooling ne peut trouver qu'un sous-ensemble. Les approches pour aider à surmonter ce problème incluent de compléter la liste des jugements de pertinence avec des documents pertinents supplémentaires découverts manuellement au cours des résultats des futures recherches.

Diverses techniques et approches ont été proposées dans la littérature pour rendre le processus d'évaluation de la pertinence plus efficace, mais elles se basent toujours sur les principes de ces deux méthodes citées ci-dessus. Par exemple, la méthode proposée par (Soboroff & Robertson, 2003) qui emploie aussi la stratégie de pooling ou la technique présentée par (Sanderson & Joho, 2004) qui utilise qu'un seul système de RI, puis ils proposent de valider les K premiers documents de résultat autant que des documents pertinents pour faire une expansion de la requête et recommencer à nouveau la recherche (la stratégie de pooling avec un seul système). Ce processus peut être refait autant de fois afin d'améliorer la qualité. Un autre algorithme proposé par (Carterette, Allan, & Sitaraman, 2006) basé sur la comparaison par paires de systèmes capables d'effectuer une corrélation de rang élevé avec un très petit ensemble de jugements. Ce qui fait uniquement les documents communs qui apparaissent plus dans leurs listes de résultat seront jugés.

D'autres travaux (Oard & Webber, 2013; Yilmaz, Kanoulas, & Aslam, 2008) ont introduit une autre méthode basée sur un échantillonnage stratifié dans lequel les documents d'un pool sont divisés en strates et échantillonnés de manière à optimiser l'effort de jugement et à mieux estimer les mesures d'efficacité. Ils ont introduit ce qu'ils ont appelé la précision moyenne inférée étendue.

En réalité, la génération de l'évaluation de la pertinence est très couteuse, elle demande souvent beaucoup de temps et de travail. Cela conduit souvent à un goulot d'étranglement dans la création de collections de test. Les problèmes pratiques qui se posent souvent lors des évaluations de la pertinence comprennent:

- **Qui devrait faire les évaluations?** Ce sont les évaluateurs ou les juges de pertinence. Idéalement, le générateur des topics effectue également des jugements de pertinence. Les auteurs dans (Bailey et al., 2008) ont comparé les évaluations de la pertinence des juges qui étaient des experts dans le domaine et d'autres qui ne l'étaient pas. Ils ont constaté de différents résultats de tests ce qui avaient affecté sur les mesures de l'efficacité du système (bien que le classement général soit resté le même).
- **Que doivent faire les évaluateurs?** Le principe de l'évaluation de la pertinence doit être déterminé à l'avance et des orientations claires doivent être données aux évaluateurs sur la manière de procéder à une évaluation de la pertinence. Le travail de (Kinney et al., 2008) a montré que les évaluations de pertinence étaient obtenues de manière plus fiable si des descriptions détaillées ont été données de ce que l'utilisateur recherchait.

- **Combien d'évaluations faut-il faire?** Les résultats expérimentaux dans (Carterette, Pavlu, Kanoulas, Aslam, & Allan, 2008) ont montré et suggéré qu'il vaut mieux évaluer moins de documents pour plus de topics (superficiels et larges) que de faire des jugements exhaustifs pour moins de topics (étroits et profonds).
- **Qu'en est-il de la recherche de documents pertinents manquants?** Si un nouveau système ou une nouvelle configuration trouve des documents pertinents qui n'ont pas été identifiés auparavant alors les performances ne sont pas maximales (en particulier pour la recherche ad hoc et l'utilisation de techniques de Pooling). En conséquence, les évaluateurs doivent utiliser des mesures de l'efficacité du système qui peuvent traiter des jugements incomplets (Aslam, Pavlu, & Yilmaz, 2006), ou les évaluateurs doivent générer de nouveaux jugements pour les résultats obtenus à partir des nouveaux systèmes.

I.5.2.3. Topics (Sujets)

Dans la recherche d'information ad hoc, la collection de test doit contenir un ensemble d'énoncés décrivant les besoins d'information des utilisateurs typiques. Celles-ci peuvent être exprimées sous forme de requêtes soumises au SRI, de questions ou de descriptions écrites plus longues. Par exemple, TREC utilise la notion de Topic (sujet), qui se compose généralement de trois champs: une requête, un titre et une description. Le champ de requête représente un ensemble typique de mots-clés qu'un utilisateur peut émettre pour un topic donné. Le champ de titre fournit une description plus longue par une phrase qui s'accorde au besoin de l'information de l'utilisateur. Le champ de description décrit plus en détail les informations de ce que nous cherchons. Les problèmes pratiques qui surviennent souvent lors de la création des topics sont:

- **Comment générer un ensemble approprié de topics?** Dans l'idéal, des requêtes réalistes devraient être développées pour la collection de test. Celles-ci pourraient être créées par des experts ou récupérées depuis des exemples réels des requêtes écrites par les utilisateurs.
- **Combien de topics sont nécessaires pour obtenir des résultats d'évaluation fiables?** Dans la plupart des travaux réalisés, un minimum de 25 topics doit être inclus dans la collection de test pour garantir sa fiabilité. Cependant, les résultats du travail de (Carterette et al., 2008) ont montré qu'avec un nombre de topics réduits mais avec un grand nombre de requêtes conduit à une évaluation plus fiable.
- **Les topics représentent-ils un ensemble suffisamment diversifié de besoins d'information?** Souvent, les requêtes sur un type de topic doivent être sélectionnées avec des caractéristiques variables pour tester les SRI sous plusieurs paramètres. Par exemple, inclure des requêtes plus courtes et plus longues, des requêtes pour des entités spécifiques (par exemple, des personnes ou des lieux) par rapport à des requêtes plus basées sur le sujet du topic. La recherche a montré que certaines requêtes sont plus utiles dans l'évaluation que d'autres (Mizzaro & Robertson, 2007).

- **Comment les topics devraient-ils être exprimés?** Les topics peuvent être exprimés sous la forme d'un ensemble de mots-clés et / ou de descriptions verbales. Pour d'autres formes de médias, cela peut inclure des descriptions non textuelles. Par exemple, pour la recherche d'images, les topics peuvent inclure des exemples d'images pouvant être utilisées pour lancer la recherche. Il convient toutefois de noter que les requêtes réalistes dans la pratique ont tendance à être brèves et ambiguës.

I.5.2.4. Requêtes

Il est nécessaire de créer les listes des requêtes selon plusieurs topics (sujets), de différentes tailles et avec de vocabulaire plus ou moins lié au domaine de la collection, afin d'imiter étroitement les requêtes réelles des utilisateurs. Un topic peut avoir plusieurs requêtes. Certains travaux proposent même de créer des requêtes contenant des fautes d'orthographe surtout pour les collections de recherche pour internet. Les moteurs de recherche commerciaux du web tels que Google (www.google.com) et Bing (www.bing.com) sont généralement évalués à l'aide de requêtes réelles issues des journaux de requêtes (Spink, 2012). Pour le nombre de requêtes proposées pour l'évaluation, Voorhees suggère et estime que le nombre suffisants est de 25 requêtes (Voorhees, 2000). Par contre le travail de (Sanderson & Joho, 2004) a conclu dans leur technique proposée qu'il faut un nombre plus de 25 requêtes pour mieux évaluer l'efficacité d'un système de RI.

I.6. Conférences et Forums pour les compagnes d'évaluation des SRIs

Cette section présente le standard d'évaluations et les différents aspects des conférences et des campagnes d'évaluation qui se déroulent annuellement ou bi-annuellement telles que TREC et CLEF, dans lesquelles, nous présentons leurs objectifs et la manière dont les pistes d'évaluations sont construites, les différentes tâches des systèmes de RI et les collections de test associées. Les tâches des campagnes d'évaluation couvrent la recherche d'information, la recherche spécifique, le filtrage, la recherche multilingue, la détection et le suivi des topics, et les systèmes Question/Réponse.

I.6.1. Standard et protocole d'évaluations

Le but des expériences de *Cranfield* était de créer une situation de type laboratoire où, libérée autant que possible de la combinaison de variables opérationnelles, les performances des langages d'indexation pourraient être considérées isolément "*a laboratory type situation where, freed as far as possible from the combination of operational variables, the performance of index languages could be considered in isolation*" (Cleverdon, 1967). Cette approche a défini le besoin d'une collection commune de documents, des taches de requêtes et des mesures afin d'évaluer différentes stratégies d'indexation dans des conditions contrôlées de l'environnement opérationnel.

La manière la plus courante pour utiliser l'approche de *Cranfield* est de comparer diverses stratégies de recherche et des systèmes, ce qui est référée à l'évaluation

comparative. L'idée est de mettre l'accent sur la performance relative entre les systèmes, plutôt que sur les scores absolus d'efficacité du système. L'évaluation des performances à l'aide de l'approche de *Cranfield* se fait sur une collection de test et elle nécessite principalement les étapes suivantes:

1. Sélectionner les différentes stratégies ou les différents systèmes de RI à comparer;
2. Lancer la recherche pour créer des listes classées de documents (appelées Runs) pour chaque requête (liée à un topic);
3. Calculer l'efficacité de chaque résultat de recherche pour chaque requête de la collection de test, par rapport aux documents pertinents avec des mesures basées sur la pertinence (par exemple, la précision et le rappel) ;
4. Faire la moyenne des scores sur toutes les requêtes pour calculer l'efficacité globale de la stratégie ou du système;
5. Utiliser les scores pour déterminer la meilleure approche et de classer les stratégies et les systèmes à tester les uns par rapport aux autres. En outre, des valeurs de pourcentage peuvent être utilisées pour déterminer si les différences entre les scores d'efficacité des stratégies ou des systèmes et leurs classements sont significatives.

I.6.2. Conférences et Forums d'évaluation

Après les premiers tests expérimentaux des systèmes de RI, tel que SMART développé par Salton (Gerard Salton, 1971a, 1971b) dans les années soixante-dix (1970), peu de corpus et des collections limitées ont été construits pour l'expérimentation et l'évaluation des nouveaux algorithmes et méthodes proposés jusqu'au début des années quatre-vingt-dix (1990). Quant aux premiers tests ont été réalisés sous des compagnes d'évaluation menées par la première conférence internationale d'évaluation, appelée *Text REtrieval Conference* (TREC). Ensuite, plusieurs autres campagnes d'évaluation et des conférences telles que *Conference and Labs of the Evaluation Forum* (CLEF), *Forum for Information Retrieval Evaluation* (FIRE), *Initiative for the Evaluation of XML Retrieval* (INEX), *Document Understanding Conferences* (DUC) et NII (*National Institute of Informatics*) *Test Collection for Information Resources* (NTCIR) ont été créées et mises en place.

Depuis alors, ces campagnes d'évaluation ont contribué au développement de la recherche scientifique dans le domaine de la RI de manière significative et rapide, car elles visaient à offrir des ateliers de tests et d'évaluations de différents algorithmes, des modèles et des techniques pour combler les difficultés auxquelles les chercheurs étaient confrontés. Ces campagnes et forums d'évaluation ont conduit à:

1. Création et standardisation des collections de test pour permettre la comparaison entre les différents nouveaux modèles et algorithmes proposés.
2. Offrir et unifier des pistes et des collections standards pour évaluer et comparer l'efficacité entre les différents systèmes de RI.

3. Organisation des conférences et des ateliers réguliers pour permettre aux participants (des chercheurs et développeurs du domaine) de présenter leurs nouvelles approches et techniques.
4. Publication et l'adoption des nouveaux algorithmes pour aider au développement ultérieur du domaine de RI.

I.6.2.1. TREC

Text **RE**trieval **C**onference (TREC[1]) a été lancé par l'Institut national des normes et de la technologie (the National Institute of Standards and Technology[2] NIST), le NIST promeut l'innovation et la compétitivité industrielle aux États-Unis en faisant progresser la science, les normes et la technologie de mesure de manière à améliorer l'économie. Les conférences TREC ont été débutées dans les années quatre-vingt-dix pour soutenir la collaboration et le transfert de technologie entre les universités, l'industrie et le gouvernement dans le domaine de la recherche d'information textuelle.

TREC a expérimenté depuis 1992 une vaste série d'évaluations et d'essais pour des systèmes de IR. Au début de ces séries de tests, il y avait plusieurs pistes de différentes tailles de collections, la piste TREC recherche Ad hoc était la plus connu, elle comprenait des collections de test sur 6 CD contenant 1,89 million de documents (principalement, mais pas exclusivement, des articles de presse (newswire articles) et des jugements de pertinence pour 450 besoins d'information, qui sont appelés sujets (topics) exprimés par un passage de texte (requêtes textuelle). Les collections de test individuelles sont définies sur différents sous-ensembles de ces données. Les premiers TREC comprenaient chacun 50 besoins d'information, évalués sur des ensembles de différents documents. Par la suite, Les TREC6 à TREC8 fournissaient 150 besoins d'information sur environ 528 000 articles d'actualités de presse et des articles du service d'information sur la radiodiffusion étrangère. C'était à l'époque la meilleure sous-collection à utiliser dans les travaux futurs, car c'était la plus grande et les sujets (topics) sont les plus cohérents. Étant donné que les collections de documents de test sont si grandes et il n'y avait pas de jugements de pertinence exhaustifs.

A partir des années 2001, TREC a créée plusieurs collections de RI en ajoutant d'autres langues à savoir l'arabe et le chinois sous d'autre formats comme le vocal, l'XML, des documents de blogs et des pages web. TREC offre aussi d'autre pistes pour le filtrage des données, les systèmes Question/Réponse, etc.

I.6.2.2. CLEF

CLEF[3] (The **C**onference and **L**abs of the **E**valuation **F**orum), La conférence et les laboratoires du forum d'évaluation est une conférence d'évaluation qui promeut la recherche et le développement du domaine de la recherche d'information dans le contexte multilingue. La première campagne d'évaluation CLEF a eu lieu en 2000 et comportait trois volets d'évaluation: multilingue, bilingue et monolingue (non anglais) fonctionnant sur les langues

[1] https://trec.nist.gov/
[2] https://www.nist.gov/
[3] http://www.clef-initiative.eu/

européennes. Depuis lors, le CLEF a continué à impliquer de nouvelles langues et de nouvelles tâches spécifiques dans le domaine de la recherche d'information multilingue. CLEF promeut la recherche et le développement en fournissant une infrastructure pour:

- Tester et évaluer des systèmes multilingues et multimodaux.
- Étudier l'utilisation de données non structurées, semi-structurées, hautement structurées et sémantiquement enrichies.
- Explorer des nouvelles techniques d'évaluation et des méthodes d'utilisation des données expérimentales.
- Créer de collections de test pour l'évaluation et la comparaison entre les systèmes.
- Discuter des résultats, comparer des approches, échanger d'idées, partager et transférer des connaissances.

CLEF est organisée en deux étapes principales:

i. Une série de laboratoires d'évaluation, c'est-à-dire des laboratoires chargés d'évaluer les systèmes d'accès à l'information et des ateliers pour discuter et piloter des activités d'évaluation innovantes;
ii. Une conférence à comité de lecture sur un large éventail de questions, y compris :
    - Enquête poursuivant les activités des laboratoires d'évaluation.
    - Des expériences utilisant des données multilingues et multimodales.
    - Étude sur les méthodologies d'évaluation et les challenges.

I.6.2.3. SIGIR

SIGIR[4] (The ACM **S**pecial **I**nterest **G**roup on **I**nformation **R**etrieval), Le groupe d'intérêt spécial ACM sur la recherche d'information est une conférence internationale depuis 1978, appartient à l'un des groupes d'intérêt de ACM. ACM[5] (**A**ssociation for **C**omputing **M**achinery) est une société savante internationale pour l'informatique de membres professionnels à but non lucratif basée aux États-Unis, son siège est à New York. Elle a été fondée en 1947 et elle est la plus grande société informatique scientifique et éducative au monde, elle revendique près de 100 000 membres chercheurs, étudiants et professionnels en 2019. ACM détient des conseils en Europe, en Inde et en Chine, favorisant les opportunités de réseautage qui renforcent les liens au sein et entre les pays et les communautés techniques.

SIGIR accueille les membres de la communauté de la RI travaillant dans tous les aspects du stockage, de la recherche et de la diffusion d'information, y compris l'évaluation, l'éducation, la recherche et le développement. Les compagnes de SIGIR couvrent un large éventail de données non structurées, y compris le texte, les images, la vidéo, l'audio et la parole. Les forums de SIGIR s'intéressent en particulier à la théorie de la recherche d'information, le test et l'évaluation des nouveaux algorithmes, nouvelles approches et des nouveaux systèmes de RI.

[4] https://sigir.org/
[5] https://dl.acm.org/

La conférence de SIGIR parraine sa propre conférence et co-parraine plusieurs d'autres conférences et ateliers notamment CIKM, JCDL, WSDM, ICTIR, CHIIR, AFIRM, elle a récemment lancé le programme *SIGIR Student Travel Grant* pour aider les étudiants à assister à ses conférences, elle publie *SIGIR Forum*, un bulletin d'information semestriel et elle soutient aussi les événements liés aux RI par le biais de son programme *Friends of SIGIR*. De plus, SIGIR participe à l'initiative de *National Science Foundation*[6] visant à créer une bibliothèque nationale électronique de science, d'ingénierie et de technologie.

I.6.2.4. NTCIR

NII **T**est **C**ollections for **IR** Systems (NTCIR[7]). Depuis 1997, Le NTCIR a construit diverses collections de test de tailles similaires aux collections de TREC, en se concentrant sur la recherche d'information multilingue pour les langues de l'Asie de l'Est, où les requêtes sont créées en monolingue sur des collections de documents contenant des documents en mono et en multilingues.

Le projet NTCIR a aussi encouragé les efforts de recherche pour améliorer les technologies d'accès à l'information telles que les techniques de recherche d'information, système de résumé automatique de texte, extraction d'information et les systèmes de Question/Réponse. NTCIR a ainsi pour objectif de:

- Offrir une infrastructure de recherche qui permet aux chercheurs de mener une évaluation à grande échelle des technologies pour l'accès à l'information.
- Former une communauté des chercheurs dans laquelle les résultats basés sur des tests expérimentaux comparables sont partagés et échangés.
- Développer des méthodologies d'évaluation et des mesures de performance des technologies pour l'accès à l'information.

Récemment NTCIR a inclut dans ses forums d'autres sujets (topics), tels que, intention de recherche et exploration de tâches, traitement du langage naturel pour le domaine médical et les documents cliniques, accès mobile aux informations, requête vocale sur des documents de paroles, accès à l'information temporelle, recherche d'information mathématiques, l'analyse des tâche Lifelog (journal/carnet de vie pour une personne) et conversation en langage naturel entre l'homme et l'ordinateur STCT (*Short Text Conversation Task*).

I.6.2.5. FIRE

FIRE[8] (**F**orum for **I**nformation **R**etrieval **E**valuation), Forum pour l'évaluation de la recherche d'information a été lancé en 12 décembre 2008 dans le but de créer un atelier sud-asiatique pour des compagnes d'évaluation dans le domaine de la recherche d'information. La diversité linguistique du sous-continent indien est similaire à celle trouvée en Europe. Géographiquement, le sous-continent indien comprend six pays, à savoir le Pakistan, le Bangladesh, le Népal, le Sri Lanka, le Bhoutan et l'Inde. La population totale de cette partie

[6] https://www.nsf.gov/
[7] http://research.nii.ac.jp/ntcir/index-en.html
[8] http://fire.irsi.res.in/

du monde est d'environ 1 800 millions d'habitants et environ 25 langues officielles sont utilisées par cette population. Parmi les principales langues de cette région, l'hindi et le bengali figurent parmi les dix langues les plus parlées au monde.

La conférence FIRE est homologue à celles des ateliers d'évaluation de TREC, CLEF et NTCIR, qui aident la recherche et le développement en fournissant des collections de test standard réutilisables et des forums communs pour comparer les modèles et les techniques. Au début FIRE avait les objectifs suivants:

- Encourager la recherche sur les technologies d'accès à l'information en langue indienne en fournissant des collections de test réutilisables à grande échelle pour l'expérimentation de la RI pour la langue indienne.
- Fournir une infrastructure d'évaluation commune pour comparer les performances de différents systèmes de RI.
- Étudier les méthodes d'évaluation des techniques d'accès à l'information et des méthodes de construction des collections de données réutilisables à grande échelle pour l'expérimentation des SRIs.

Le forum FIRE a depuis évolué continuellement pour répondre aux nouveaux défis de l'accès multilingue à l'information. Il s'est élargi pour inclure de nouveaux domaines et sujets (topics) tels que le traitement du langage naturel, l'interaction homme-machine, la linguistique informatique, le web sémantique, l'analyse des réseaux sociaux, la détection du plagiat, l'accès aux informations juridiques, la recherche d'information de scripts mixtes et la recherche de la parole vocale.

I.6.2.6. INEX

The **IN**itiative for the **E**valuation of **X**ML Retrieval (INEX[9]), L'initiative pour l'évaluation de la recherche (des documents) XML. C'est une campagne internationale débutait en 2002, où elle fournit un espace de comparaison sous la forme de vastes collections de test et des méthodes appropriées pour évaluer l'efficacité des systèmes de RI orientés contenu sous format XML.

L'évaluation est effectuée à l'aide des collections de test assemblées spécifiquement pour évaluer des tâches de recherches particulières liées aux documents XML. Une collection de test se compose d'un corpus de documents, d'un ensemble de besoins des utilisateurs (c'est-à-dire des sujets – topics-) et les jugements de pertinence. Les caractéristiques des collections de test traditionnelles ont été ajustées pour évaluer de manière appropriée l'efficacité de la recherche XML: la collection de documents comprend des documents balisés en XML, les topics spécifient des requêtes relatives à la fois au contenu et à la structure, et les évaluations de la pertinence sont effectuées au niveau des éléments. En outre, la pertinence est mesurée de manière à quantifier de façon appropriée la capacité du système à renvoyer la granularité correcte des éléments XML. Ainsi, les systèmes de recherche XML visent à exploiter la structure logique des documents pour

[9] https://www.is.inf.uni-due.de/projects/inex/index.html.en

trouver des composants dans le document (c'est-à-dire des éléments XML) en réponse à la requête d'un utilisateur, plutôt que des documents entiers.

Les premières éditions jusqu'en 2004, la collection de documents comprenait des articles, balisés en XML, provenant des revues de publication de l'*IEEE Computer Society*, où chaque document contient plus de mille éléments XML, (en moyenne, un article contient 1 532 nœuds XML). En 2005, la collection est enrichie pour 11 millions d'éléments XML. Dans la période (2006-2007) d'autres documents du projet Wikipedia sont ajoutés totalisant plus de 60 Go (4,6 Go sans images) et 30 millions d'éléments XML. Ensuite des collections des livres numérisés balisés en XML ont été ajoutées aux pistes d'évaluations en 2009.

I.6.2.7. DUC / TAC

Document **U**nderstanding **C**onferences (DUC[10]) – Conférences sur la compréhension des documents, ce sont des séries d'ateliers consacrées à l'évaluation des performances des systèmes de synthèse ou de résumer du texte, entre la période de 2001-2007. En 2008, DUC est devenu une piste dans le cadre des conférences de (TAC[11]) *the **T**ext **A**nalysis **C**onference* Conférence d'analyse de texte.

TAC est une série d'ateliers d'évaluation organisés pour encourager la recherche sur le traitement du langage naturel et les applications connexes, en fournissant une vaste de collections de test, des procédures d'évaluation communes et un forum permettant aux participants de partager leurs résultats. Dans lequel, le TAC comprend des ensembles de tâches appelées «pistes», dont chacune se concentre sur un sous-problème particulier du TAL y compris la recherche d'information et les systèmes de question/réponse.

I.6.2.8. Autres conférences

Ils existent d'autres conférences équivalentes reconnues et liées aux domaine de la RI telles que :

ECIR[12] (The annual **E**uropean **C**onference on **I**nformation **R**etrieval), La Conférence européenne annuelle sur la recherche d'information, c'est le principal forum européen pour la présentation des nouveaux résultats de recherche dans le domaine de la recherche d'information.

AIRS[13] (**A**sia **I**nformation **R**etrieval **S**ocieties), la conférence des sociétés de recherche d'information en Asie vise à rassembler des chercheurs et des développeurs pour échanger de nouvelles idées et les dernières réalisations dans le domaine de la recherche d'information. Le scope de la conférence couvre les aspects théoriques, les systèmes, les applications, les technologies de la RI pour les données textuels, les images, les vidéos, l'audio et les multimédias.

ADCS (**A**ustralasian **D**ocument **C**omputing **S**ymposium), le symposium Australasien (la Nouvelle-Zélande et l'Australie) sur l'informatique des documents depuis

[10] https://duc.nist.gov/
[11] https://tac.nist.gov//
[12] https://irsg.bcs.org/ecir.php
[13] http://airs2019.ouhk.edu.hk/

2012 sponsorisé par ACM[14], il couvre la recherche académique et commerciale dans la recherche d'information et la gestion documentaire.

DESIRES[15] (**D**esign of **E**xperimental **S**earch & **I**nformation **R**etrieval **S**ystems), Le design des systèmes expérimentaux de recherche et d'extraction d'information est une conférence biannuelle axée sur les aspects technologiques innovants des systèmes de recherche et d'extraction d'information.

I.7. Conclusion

Nous avons présenté dans ce chapitre l'état de l'art du domaine de la recherche d'information. Nous avons abordé les notions et concepts de base de la recherche d'information, les différents principaux composants d'un système de recherche d'information ainsi que son évaluation à travers les diverses mesures adaptées à savoir la précision, le rappel, la F-mesure, la précision interpolée, … etc. La conception et la construction des collections de test des systèmes de RI ont été également exposées, ainsi que les différentes conférences et forums internationaux les plus réputées ont été détaillés à travers leurs principes d'activités, leurs objectifs et leurs propriétaires ainsi nous avons indiqué et donné leurs liens d'accès sur le web. Dans le prochain chapitre, nous présentons les approches sémantiques récentes ainsi que les différentes techniques réussies qui sont utilisées pour analyser les textes afin d'améliorer les performances de la recherche d'information.

[14] https://dl.acm.org/
[15] http://desires.dei.unipd.it/

Chapitre II. Sémantique dans les textes et la recherche d'information sémantique

II.1. Introduction

Dans ce deuxième chapitre, nous présentons d'abord la sémantique du texte, par la définition du sens des mots, la définition des relations sémantiques telles que la synonymie, la similarité des mots, l'antonymie, l'hyperonymie, la méronymie, aussi les différentes techniques de la désambiguïsation du sens des mots (WSD) et ainsi la similarité sémantique des phrases. Ensuite, nous détaillons la ressource sémantique du WordNet. Puis, nous exposons également les récents modèles qui utilisent la sémantique vectorielle à savoir ; le Word2vec, le GloVe, le FastText, l'ELMo, le Bert et l'openAI GPT. A la fin du chapitre, nous présentons l'état de l'art sur les différents travaux qui ont abordé la problématique de la recherche d'information sémantique et ses différentes méthodes pour l'indexation sémantique et la reformulation sémantique des requêtes.

II.2. Recherche d'information et traitement automatique de la langue

La Recherche d'Information (RI) est généralement attribuée à Gerard Salton (Salton, 1968), professeur d'informatique à l'université Cornell[1] qui, dans les années soixante (1960), a créé le premier groupe de recherche dédié à la recherche d'information. Dès le départ, la recherche d'information a été marquée par une rivalité avec une autre jeune discipline, l'Intelligence Artificielle (IA) (Alessandro Moschitti[2]).

L'une des disciplines émergeant par l'IA est bien le Traitement Automatique de la Langue (TAL), mais il y avait une différence fondamentale dans leurs approches techniques, la RI était fondée plus sur les statistiques et les formules mathématiques, tandis que l'IA et le TAL étaient plutôt basés sur la logique et les règles. Et pourtant, il y avait un chevauchement entre le TAL, qui cherchait à traiter les besoins de l'utilisateur présentés par des requêtes formulées en langage naturel et la recherche d'information, qui indexait, sélectionnait ou classait automatiquement les résultats de recherche en fonction de leurs

[1] https://www.cornell.edu/
[2] http://disi.unitn.it/~moschitti/ un scientifique principal de l'organisation Alexa Intelligence Artificielle et le groupe SIGIR.

contenus. Mais à ce stade, le TAL reposait toujours principalement sur des systèmes basés sur des règles et des techniques linguistiques, tandis que la RI avait continué à développer des méthodes statistiques et probabilistes plus efficaces.

Dans ce temps, des chercheurs dans le domaine de TAL ont essayé de développer des moteurs de recherche sémantique en analysant le contenu des requêtes et les documents par les techniques de l'analyse morpho-lexicale, l'analyse syntaxique et la reconnaissance des entités nommées, par contre les chercheurs de RI ont développé des modèles plus efficaces en se basant uniquement sur les racines des mots et des calculs basés sur des équations de fondement mathématiques. En conséquence les premiers systèmes de RI n'avait pas besoin des approches avancées de TAL afin de réaliser des performances en sélectionnant les documents pertinents pour des requêtes données.

En revanche, les systèmes de RI modernes se sont rapprochés plus à la discipline du TAL grâce à l'utilisation de ses nouvelles techniques et ses récentes applications, puisque ces systèmes de RI ne sélectionnent plus simplement des liens vers des pages web ou des documents mais ils récupèrent et ils sélectionnent des informations adéquates extraites des pages web ou des documents qui sont étiquetés en fonction du type de contenu, ou des extraits de documents que les utilisateurs sont susceptibles de les trouver utiles. Les résultats de recherche de ces nouveaux systèmes de RI sont donc une sorte d'hybridation ou de production d'informations qui ont vraiment besoin des techniques de TAL, à savoir ; l'extraction d'information, l'analyse des sentiments, l'étiquetage sémantiques, la classification des documents, etc. De plus, ces techniques utilisent davantage des ressources langagières sémantiques comme les dictionnaires, les ontologies ou les corpus annotés.

Très récemment avec la révolution des résultats par les algorithmes de machine learning et deep learning, les implications des techniques de TAL ont engendré des nouvelles approches basées sur le plongement de mots (Word-Embedding) pour représenter des mots ou des séquences de mots comme des points dans un espace vectoriel, par lequel un mot est représenté par un vecteur de valeurs numériques et toute l'interprétation sémantique et les calculs de similarités sont réalisés en utilisant ces vecteurs de Word-Embedding (par exemple l'angle entre deux vecteurs représente le degré de similarité sémantique entre la requêtes et le document). En effet ; la communauté de RI a apprécié ces nouvelles approches de Word-Embedding et la RI débute à bénéficier de leurs techniques dans les différentes phases du SRI à savoir ; l'indexation, la reformulation des requêtes, les calculs des pertinences et l'appariement.

Finalement, le TAL et la RI sont maintenant encore plus proches, puisqu'ils se basent sur presque les mêmes formules et ils utilisent les mêmes outils, d'où la majorité des travaux récents de la RI pour les textes chevauchent avec les travaux du TAL.

II.3. Sémantique du texte

II.3.1. Sens du mot

Un sens (ou sens du mot) est une représentation discrète d'un aspect sémantique d'un mot, c'est-à-dire, une représentation mentale d'une chose exprimée par un mot. Suivant les principes de la lexicographie, il est plus compréhensible de représenter chaque sens avec une *définition et/ou un exemple* lié à un contexte. Pour comprendre ce principe, nous donnons les exemples suivants pour les mots 'souris' et 'barrage' pris du dictionnaire.

Souris sens1 : Une souris est un périphérique d'entrée pour un système informatique.

sens2 : Un petit animal rongeur.

Barrage : sens1 : Action de barrer un passage ou faire obstacle sur une voie.

sens2 : Ouvrage barrant un cours d'eau pour faire une retenue.

Les mots de la langue sont alors ambigus, par lequel, le même mot peut être utilisé pour signifier différentes choses, dans l'exemple du mot 'souris', il a au moins deux sens: (1) un petit rongeur, ou (2) un dispositif manuel pour contrôler le curseur sur un écran. Le mot 'barrage' peut signifier : (1) un ouvrage d'art pour stocker de l'eau, ou (2) un barrage de contrôle de police sur la route.

Nous disons que les mots 'souris' ou 'barrage' sont polysémiques « avoir plusieurs sens » (polysémie est un mot grec composée par deux parties : *poly* : plusieurs + *sema* : signe, marque).

Connaître la relation entre deux sens peut jouer un rôle important dans la compréhension du langage. Considérons la relation d'antonymie, deux mots sont des antonymes s'ils ont des sens opposés, comme long et court, ou haut et bas, dont la distinction est très importante entre les sens de ces mots (si un utilisateur demande à un agent-robot de baisser la vitesse, il serait dramatique de l'augmenter à la place).

L'une des solutions proposées dans la littérature est la désambiguïsation du sens des mots (WSD : Word Sense Disambiguation), c'est la tâche de déterminer quel sens d'un mot est utilisé dans un contexte particulier et d'identifier les relations sémantiques exactes entre les sens des mots, cette technique est longuement utilisée dans la linguistique informatique et de nombreuses applications liées au TAL.

II.3.2. Relations sémantiques

Nous présentons dans cette section les relations entre les sens des mots, en particulier celles qui sont utilisées dans le TAL et la RI et qui ont déjà fait l'objet des études dans la sémantique des textes à savoir, la synonymie, l'antonymie et l'hypernymie.

II.3.2.1. Synonymie

Le principe est que lorsque les sens de deux mots (lemmes) différents sont identiques, ou presque identiques, nous disons alors que les deux sens de ces deux mots sont **synonymes**. Des exemples de synonymes incluent des paires de mots tels que :

voiture/automobile,أريكة/ كنبة (canapé en français), filbert/ hazelnut (noisette en français)

Une définition plus formelle de la synonymie (entre les mots plutôt que les sens) est que deux mots sont synonymes s'ils sont substituables l'un à l'autre dans une phrase sans changer les *conditions de vérité* de la phrase, les situations dans laquelle la phrase serait vraie. Nous disons souvent dans ce cas que les deux mots ont le même *sens propositionnel.* Bien que les substitutions entre certaines paires de mots comme voiture / automobile ou eau / H_2O préservent la vérité des phrases, tandis que les mots n'ont toujours pas le même sens identique. En effet, il est probable qu'il n'y a pas deux mots absolument identiques dans leur sens. L'un des principes fondamentaux de la sémantique, appelé *principe de contraste* (Clark & MacWhinney, 1987), qui affirme qu'une différence de forme linguistique est toujours associée à une différence de sens. Par exemple, les mots H_2O/NaCl sont utilisés dans des contextes scientifiques et ça serait inapproprié dans un contexte habituel (dans un texte généraliste pour un publique non spécialiste, l'utilisation des mots *'eau/sel'* serait plus appropriée) et cette différence de genre fait partie du sens du mot. En pratique, le mot synonyme est donc utilisé pour décrire une relation de synonymie approximative ou brute.

Mais de plus, la synonymie est en fait une relation entre les sens plutôt que les mots. Considérant les mots '**big'** et '**large'** (grand). Ceux-ci peuvent sembler être des synonymes dans les phrases suivantes, puisque nous pourrions échanger les mots '**big'** et '**large'** en conservant le même sens dans chacune de ces phrases (p1 et p2):

- **p1**:"How **big** is that bus?" (Quelle est la taille de ce bus?)
  Après substitution, p1 devient : "How **large** is that bus?" (Quelle est la taille de ce bus?)
- **p2**:"Would we move on a **large** or small bus?" (Déplacerions-nous dans un grand ou un petit bus?)
  Après substitution, p2 devient : "Would we move on a big or small bus?" (Déplacerions-nous dans un grand ou un petit bus?)

Cependant, dans les phrases suivantes, nous ne pouvons pas remplacer le mot '**big'** par le mot '**large'.**

- "Miss Zahra, for instance, became a kind of **big** sister to Karim."
- "Miss Zahra, for instance, became a kind of **large** sister to Karim."

« Mademoiselle Zahra, par exemple, est devenue une sorte de grande sœur de Karim. »

Ceux-ci est parce que le mot '**big'** a un sens qui signifie être plus âgé, alors que '**large'** n'a pas ce sens. Ainsi, nous disons que certains sens du **'big'** et du **'large'** sont (presque) synonymes alors que d'autres ne le sont pas.

II.3.2.2. Similarité des mots

Bien que les mots n'aient pas beaucoup de synonymes, la plupart des mots ont beaucoup de mots similaires. Le mot ***chat*** n'est pas synonyme du mot ***chien***, mais ***chat*** et ***chien*** sont évidemment des mots similaires. En passant de la synonymie à la similarité, il est utile de passer des relations entre les sens des mots (comme la synonymie) aux relations entre les mots (comme la similarité).

La notion de similarité de mots est très utile dans la plupart des analyses sémantiques. Le principe est de savoir à quel point deux mots sont similaires peut aider à calculer la similarité de la signification de deux expressions ou de deux phrases, cette approche est largement utilisée dans les tâches de compréhension du langage naturel telles que les systèmes de question/réponse, la traduction automatique, les systèmes de résumage automatique, …etc.

Les techniques d'avoir des valeurs pour la similarité entre les mots sont soit : automatique par les calculs en utilisant des formules mathématiques ou soit : par demander aux humains (experts) de juger à quel point un mot est similaire à un autre. Un certain nombre de travaux ont déjà réalisé des datasets avec des valeurs de similarités des mots. Par exemple, *SimLex-999* un dataset réalisé par (Hill, Reichart, & Korhonen, 2015) qui donne des valeurs sur une échelle de 0 à 10, comme il montre l'exemple d'échantillon ci-dessous. Nous remarquons qu'ils existent des mots quasi-synonymes comme *[vanish, disappear] (*disparaître en français) et d'autres qui semblent à peine avoir quelque chose en commun comme les deux mots *[hole, agreement] (*trou et accord, en français):

Mot1	Mot2	Similarité
vanish	disappear	9.8
belief	impression	5.95
muscle	bone	3.65
modest	flexible	0.98
hole	agreement	0.3

II.3.2.3. Relation de mots

Les sens de deux mots peuvent être reliés autrement que par la similarité. L'une de ces classes de connexions est appelée *relation entre les mots* (Budanitsky & Hirst, 2006), aussi traditionnellement appelée en psychologie *association de mots*.

Considérons la signification des mots ***café*** et ***tasse***. Le ***café*** n'est pas similaire à la ***tasse***; ils ne partagent pratiquement aucune caractéristique (le café est une plante ou une boisson, tandis qu'une tasse est un objet fabriqué avec une forme particulière). Mais le café et la tasse sont clairement reliés; ils s'associent en co-participant à un événement du quotidien (l'événement de boire du café dans une tasse). De la même manière, ***chirurgien*** et ***scalpel*** , ils ne sont pas similaires mais ils sont reliés de manière éventuelle (un chirurgien a tendance à utiliser un scalpel). L'une de types des relations entre ces deux mots est qu'ils appartiennent au même champ sémantique. Par définition, un champ sémantique est un ensemble de mots

qui couvrent un domaine sémantique particulier et qui entretiennent des relations structurées entre eux. Par exemple, les mots peuvent être reliés dans le domaine sémantique ; des hôpitaux (les mots : *hôpital, chirurgien, infirmière, anesthésique, scalpel, …*), des restaurants (les mots *: menu, nourriture, serveur, plat, chef, …)* ou des maisons (les mots : *famille, père, mère, chambre, porte, toit, cuisine, …*).

Les champs sémantiques sont aussi reliés à des modèles thématiques. Plusieurs techniques et algorithmes ont été proposés à base d'apprentissage automatique sur des grands corpus afin d'extraire ces ensembles de mots associés qui forment les champs sémantiques. En pratique, les champs sémantiques et les modèles thématiques sont des outils très utiles pour découvrir la structure thématique et la sémantique des textes.

Nous introduisons aussi d'autres importantes relations entre les sens telles que ; l'antonymie (l'opposé), l'hypernymie (IS-A, est-un) et la méronymie (relations partie-tout).

II.3.2.4. Antonymie

Bien que les synonymes sont des mots ayant un sens identique ou similaire, les antonymes sont des mots ayant un sens opposé, par exemples :

Mot	Antonyme	en français
بارد	ساخن	froid/chaud
up	down	haut/bas
dark	light	foncé/clair
rapide	lent	

Deux sens peuvent être des antonymes s'ils définissent une opposition binaire ou ils sont aux extrémités opposées d'une certaine échelle. C'est le cas pour *long/court*, *rapide/lent* ou *grand/petit*, qui se trouvent aux extrémités opposées de l'échelle de *longueur*, de *vitesse* ou de *taille*. Un autre groupe d'antonymes, les inversions, décrivent un changement ou un mouvement dans des directions opposées, telles que *montée/descente* ou *haut/bas*.

Les antonymes diffèrent donc complètement par rapport à un aspect de leurs sens (leurs positions sur une échelle ou leurs directions) mais ils sont par ailleurs très similaires, ils partagent presque tous les autres aspects du sens. Ce qui fait la distinction automatique les synonymes des antonymes par des algorithmes constitue un challenge et une opération difficile.

II.3.2.5. Hyperonymie / Hyponymie

Les sens des mots peuvent aussi être reliés par taxonomie de généralisation (superclasse) ou de spécification (sous-classe). Nous disons un sens de mot1 est plus général (superclasse) d'un autre sens du mot2 si la relation du sens de mot1 par rapport au sens du

mot2 est hyperonyme (appelée aussi hypernyme), par exemple le sens du mot1 *animal* est hyperonyme au sens du mot *chat*, un *fruit* est hyperonyme de *pomme*, la *couleur* est hyperonyme de *rouge*. Inversement, la relation de spécification (sous-classe) est appelée hyponymie, par exemple un *chat* est un hyponyme d'*animal*, une *pomme* est un hyponyme de *fruit* et le *rouge* est un hyponyme de *couleur.* La Figure II.1 illustre un exemple de relations *hyperonymies/hyponymies* pour les couleurs.

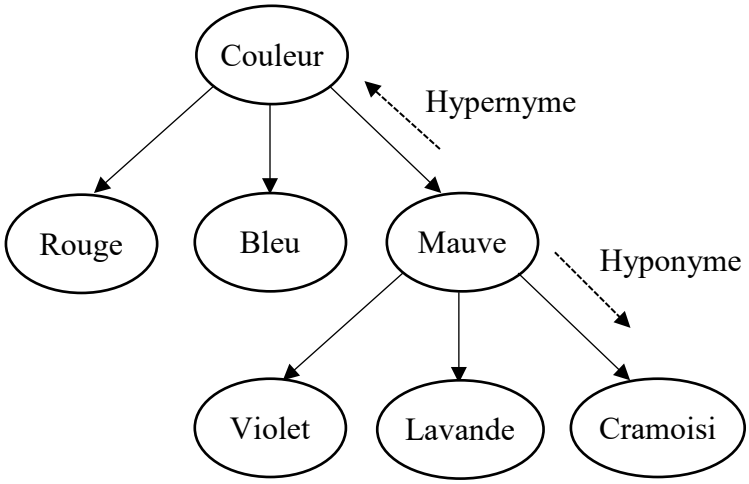

Figure II.1. Relation Hyponyme/ Hypernyme

Une autre formalisation utilisée dans la sémantique des textes pour la relation hyperonymie/hyponymie est la hiérarchie **Is-a** (est-un), dans laquelle nous disons X **Is-a** Y ou X est hyponyme de Y (ou Y est hyperonyme de X). La relation **Is-a** (hyperonymie/hyponymie) est généralement une relation transitive (si X **Is-a** Y et Y **Is-a** Z, alors X **Is-a** Z). La relation hypernymie est très utile pour des tâches telles que l'implication sémantique textuelle, système de question / réponse, enrichissement d'ontologie, etc. Par exemple, pour la question "Quels sont les caractéristiques de la tulipe ? ", nous savons de l'autre côté que "la tulipe est une **(Is-a)** plante", alors il est bien utile de donner les caractéristiques de la classe *plante* pour répondre à cette question.

II.3.2.6. Méronymie/ Holonymie

Une autre relation courante est la méronymie, c'est une relation qui désigne une partie constitutive ou un membre de quelque chose. Par exemple un bras fait partie d'un corps, alors nous disons qu'un *bras* est un méronyme de *corps*, une *pomme* est un méronyme de *pommier* et un *moteur* est un méronyme de *voiture*. Cette relation est indiquée formellement par l'expression ***Part-of*** ou ***Part-whole*** (partie-de ou partie-tout).

L'inverse de la relation de méronymie est une relation d'holonymie (*composé-de*). Notamment, nous disons qu'un *corps* est un holonyme de *bras*, un *pommier* est un holonyme de *pomme* et une *voiture* est un holonyme de *moteur*. La méronymie/holonymie est aussi une relation (partitive) hiérarchisée. La Figure II.2 illustre un exemple de relations *méronymie/holonymie* pour l'objet voiture.

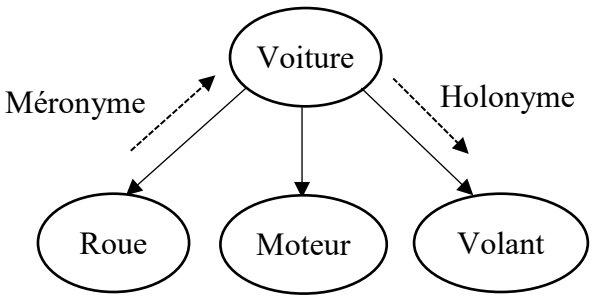

Figure II.2. Relation Méronyme/ Holonyme

II.3.3. Désambiguïsation des sens des mots (WSD)

La désambiguïsation des sens des mots (WSD : Word Sense Disambiguation) est une méthode largement utilisée pour découvrir le sens exact d'un mot ambigu dans un contexte particulier ou plus exactement de trouver le sens le plus probable d'un mot multi-sens dans une phrase (Agirre & Edmonds, 2007; Ide & Véronis, 1998; Navigli, 2009). Exemple le mot anglais "**bank"** peut avoir des sens différents comme 'institution financière', 'bord de la rivière', 'réservoir', etc. ce qui fait les mots avec plusieurs sens sont appelés mots ambigus, dont le processus de recherche du sens exact d'un mot ambigu pour un contexte particulier est appelé par *la désambiguïsation des sens des mots* (WSD). Les êtres humains ont la capacité innée de distinguer les différents sens d'un mot ambigu dans un contexte particulier, mais les machines ne peuvent fonctionner que selon les instructions. Par conséquent, différentes règles sont implémentées sur les systèmes pour effectuer ces tâches spécifiques.

La désambiguïsation des sens des mots est souvent utilisée dans la traduction automatique puisque les mots de chaque langue ont des traductions différentes en fonction des contextes de leurs utilisations, elle est également utilisée dans la recherche d'information pour résoudre l'ambiguïté dans la requête est de trouver le terme-sens exacte avant d'effectuer l'appariement de requête-index. Aussi, la WSD est utilisée dans d'autres domaines tels que l'extraction d'information (EI), le Text-Mining (fouille de texte), la reconnaissance des entités nommées et la résolution de co-référence, etc.

Il existe trois principales approches pour résoudre la problématique du WSD ; les approches basées sur les connaissances, les approches supervisées et les approches non-supervisées.

II.3.3.1. Approches du WSD basées sur les connaissances

Les approches du WSD basées sur les connaissances cherchent une représentation du sens à partir des ressources de connaissances disponibles, à savoir les dictionnaires électroniques, les thésaurus ou les bases lexicales telle que WordNet, etc. Ces approches sont établies par l'algorithme de LESK (Banerjee & Pedersen, 2002; Lesk, 1986; Moro, Raganato, & Navigli, 2014), par la similarité sémantique ou par les méthodes heuristiques (Trois types d'heuristiques sont utilisés i) le sens le plus fréquent, ii) le sens par discours et iii) le sens par collocation.

II.3.3.2. Approches du WSD supervisées

Les approches du WSD supervisées utilisent les techniques de l'apprentissage automatique (Machine Learning) à partir des données de type sens annotées manuellement (Klein, Toutanova, Ilhan, Kamvar, & Manning, 2002; Lee & Ng, 2002). Le problème est considéré comme une tâche de classification pour trouver le sens le plus probable d'un mot cible. Les sens des mots sont alors des classes et les contextes des mots forment le modèle du classificateur dont l'annotation est étiquetée manuellement à partir du dictionnaire pour former le corpus d'apprentissage. Plusieurs algorithmes sont testés à savoir, l'arbre de décision, Naïve Bayes, machine à vecteurs supports (SVM), etc. Ces approches ont démontré de bonnes performances dans les compétitions par rapport aux autres approches.

Récemment, certains modèles de réseaux de neurones profonds (Deep learning) tel que le réseau bidirectionnel de mémoire à long court terme (Bi-LSTM) a été testé pour le WSD, en utilisant le Context2vec (Melamud, Goldberger, & Dagan, 2016), dont il a montré de meilleurs résultats par rapport à d'autres méthodes supervisées.

II.3.3.3. Approches du WSD non-supervisées

Les méthodes WSD non supervisées ne dépendent ni de ressources de connaissances externes ni des corpus annotés manuellement par les sens. Ces algorithmes n'attribuent généralement pas de sens aux mots, mais ils classent (discriminent) les sens des mots sur la base d'information trouvée dans des corpus non annotés. Ces approches se distinguent en deux catégories : les approches basées sur le regroupement des contextes (context clustering) (Bartunov, Kondrashkin, Osokin, & Vetrov, 2016; Reisinger & Mooney, 2010) et les approches basées sur le regroupement des mots (word clustering) (Hope & Keller, 2013; Pantel & Lin, 2002).

II.3.3.4. Comparaison entre les approches de WSD

Le tableau II.1 suivant présente une comparaison entre les différentes approches de WSD.

Tableau II.1. Comparaison entre les approches du WSD (Pal & Saha, 2015).

Approche	Avantage	Inconvénient
WSD basée sur les connaissances	Les algorithmes de ce type donnent une plus grande précision.	Les algorithmes dépendent de la qualité des ressources externes et les définitions du dictionnaire.
WSD supervisée	Ce type d'algorithmes est meilleur que les deux autres approches	Les algorithmes ne donnent pas de résultat satisfaisant pour les langues à ressources limitées de corpus annotés
WSD non-supervisée	Pas besoin de corpus annotés de sens dans cette approche	Les algorithmes sont difficiles à mettre en œuvre et les performances sont toujours inférieures à celles des deux autres approches.

II.3.4. Similarité sémantique des phrases

Le calcul de similarité des phrases est aussi l'une des solutions pour résoudre le problème de la sémantique des textes, ces techniques sont utilisées dans plusieurs applications, telles que les systèmes question/réponse, la traduction automatique, le résumé automatique des textes et la recherche d'information.

Pas mal de techniques ont été proposées pour calculer la similarité sémantique des phrases. Certaines sont basées sur les mots co-occurrents entre les phrases (Nirenburg, Domashnev, & Grannes, 1993) ou sur les dépendances syntaxiques similaires (Mandreoli, Martoglia, & Tiberio, 2002), cependant ces techniques ne prenaient pas en considération de la sémantique, tels que les mots qui ont des sens différents (la souris peut être un animal ou un dispositif de pointage relié à l'ordinateur) ou des mots synonymes tels que siège/chaise.

D'autres approches se sont appuyées sur l'utilisation de la sémantique en utilisant des ressources externes, telle que WordNet (Leacock & Chodorow, 1998) ou des corpus de textes annotées sémantiquement (Mihalcea, Corley, & Strapparava, 2006). Cependant, ces méthodes n'induisent pas un réel score de similarité sémantique. Des chercheurs ont alors proposées d'autres approches basées sur les structures syntaxiques et les informations sémantiques, appelées approches hybrides, comme celles présentées par les travaux de (Islam & Inkpen, 2008; Yuhua Li, McLean, Bandar, O'shea, & Crockett, 2006; Šarić, Glavaš, Karan, Šnajder, & Bašić, 2012) qui prennent en compte à la fois la signification et l'ordre des mots impliqués dans la structure de la phrase pour mesurer la similarité sémantique. Ces méthodes hybrides se subdivisent en trois familles, les méthodes basées sur

l'ordre des mots, les méthodes basées sur les POS-Tags[3] (Part of Speech) et les méthodes basées sur la dépendance syntaxique.

II.4. WordNet

II.4.1. Définition

WordNet est une base de données lexicales, elle est devenu l'un des dictionnaires les plus populaires pour le traitement automatique de la langue (TAL), analyse de sémantique des textes et d'autres domaines de technologies linguistiques. Cela est principalement dû à sa structure sous forme d'hiérarchie de mots, qui est beaucoup plus facile à comprendre par les ordinateurs que la forme traditionnelle d'un dictionnaire. Le premier WordNet a été introduit par Arthur Miller (Miller, 1995) du laboratoire des sciences cognitives de l'université de Princeton, ensuite il a été étendu par Christiane Fellbaum (Fellbaum, 1998) aussi à l'université de Princeton avant d'être finalement publié sous sa forme définitive sous le nom de Princeton WordNet 3.0 en 2006, et puis c'est mis à jour à la version 3.1 en 2011.

Pendant ce temps, l'intérêt et l'utilisation des wordNets ont augmenté avec de nombreux projets à travers le monde créant des nouveaux wordNets pour des langues autres que l'anglais, par lesquelles des centaines de versions ont été introduites [4], exemple, Euro WordNet pour les langues européennes, Arabic WordNet pour l'arabe, etc. Encore, d'autres projets se sont développés autour du Princeton wordNet par son extension à des informations de sentiments (SentiWordNet) (Esuli & Sebastiani, 2006), des informations encyclopédiques (Navigli & Ponzetto, 2012), des pronoms et des exclamatifs (Da Costa & Bond, 2016) et par une terminologie spécifique au domaine (McCrae, Wood, & Hicks, 2017).

L'unité principale et composante du wordNet est appelée un synset (dérivé de deux mot **syn**onym et **set**) ou un ensemble de synonymes, elle est constituée d'une liste de mots synonymes qui, dans un certain contexte, peuvent être substitués les uns aux autres, ces unités forment des nœuds qui sont hiérarchisés par un graphe dont les arêtes sont des relations sémantiques, telles que hypernyme/hyponyme et meronyme. Un mot sous forme d'un lemme peut faire partie alors de plusieurs synsets quand le mot a plusieurs sens, puisque chaque synset se réfère à un seul sens. La Figure II.3. suivante montre un exemple des synsets reliés par des relations sémantiques :

[3] Processus d'affecter un POS (un marqueur: Det, N, V, ...) de classe lexicale à chaque mot dans une phrase.
[4] http://globalwordnet.org/resources/wordnets-in-the-world/

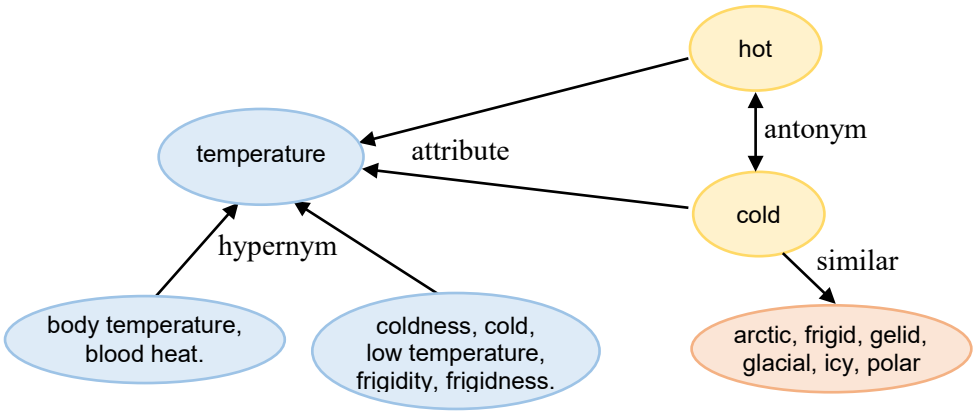

Figure II.3. Exemple graphique « WordNet » de "*temperature*"

WordNet se compose ainsi de trois bases de données distinctes, une première base pour les noms, une deuxième base pour les verbes et une troisième base pour les adjectifs et les adverbes. Chaque base de données contient un ensemble de lemmes dont chacun est annoté avec un sens ou en fait il est affecté à un synset.

Les noms sont hiérarchisés dans un arbre de relations hiérarchiques d'hypernyme (hyperonyme *Is-a*) ou de méronyme (*Part-of*) jusqu'au arriver au nom unique général de départ ***'entity'***. Les verbes sont aussi regroupés en hiérarchie, cependant il n'y a pas un verbe suprême global, pour les verbes, le graphe est plus déconnecté. Pour les adjectifs, la structure est généralement basée sur un modèle haltère où les adjectifs sont regroupés en paires d'antonymes, tels que *chaud/ froid* ou bien un sens est défini par '*en relation à'* un nom, par exemple '*français'* à '*France'*. Pour les adverbes, il y a peu de structures et de nombreux synsets d'adverbes qui n'ont aucun lien dans le graphe.

WordNet a deux types d'entités taxonomiques ; les **classes** et les **instances**. Une instance est un individu d'une classe, un nom propre qui est une entité unique, par exemple, le mot (individu) *'djanet'*[5] est une **instance** de *'town'* et *'town'* est une **classe**, et au même temps, elle est hyponyme de *'municipality'*.

Précisément, le WordNet 3.1 contient 207.272 synsets (sens), 378.203 relations, 117.798 noms, 11.529 verbes, 22.479 adjectifs et 4.481 adverbes. La moyenne des noms est de 1,23 de noms/sens et la moyenne des verbes est 2,16 verbes/sens. WordNet est disponible en ligne[6] et il est aussi téléchargeable sous plusieurs formats (OWL, XML…).

[5] Djanet (en arabe : جانت), est une commune en Algérie. C'est une oasis et elle est la principale ville du sud-est du Sahara algérien.

[6] http://wordnetweb.princeton.edu/perl/webwn

II.4.2. Synset

Dans le WordNet, le sens d'un mot est représenté par :

1) un **sysnset** exprimé par un ensemble des mots synonymes, qui peuvent être utilisés pour exprimer un concept
2) une définition,
3) parfois aussi des exemples d'utilisation.

Par exemple, le mot (lemme) "machine" appartient à la classe des noms et la classe des verbes. Plus en détail, pour le nom "machine" a 6 sens et pour le verbe "machine" a 2 sens comme présente la Figure II.4.

Nom:

1. **machine**[1] (any mechanical or electrical device that transmits or modifies energy to perform or assist in the performance of human tasks)
2. **machine**[2] (an efficient person) *"**the boxer was a magnificent fighting machine**"*
3. **machine**[3] (an intricate organization that accomplishes its goals efficiently) ***"the war machine"***
4. **machine**[4], simple machine (a device for overcoming resistance at one point by applying force at some other point)
5. **machine**[5], political machine (a group that controls the activities of a political party) ***"he was endorsed by the Democratic machine"***
6. car, auto, automobile, **machine**[5], motorcar (a motor vehicle with four wheels; usually propelled by an internal combustion engine) ***"he needs a car to get to work"***

Verbe:

7. **machine**[6] (turn, shape, mold, or otherwise finish by machinery)
8. **machine**[7] (make by machinery) "***The Americans were machining while others still hand-made cars"***

Figure II.4. Synsets du lemme *'machine'*

WordNet étiquette également chaque synset avec une catégorie lexicographique tirée d'un champ sémantique. Plus précisément, il étiquette 26 catégories de noms, 15 catégories de verbes, deux catégories d'adjectifs et une catégorie d'adverbe.

Les sysnsets incluent aussi en particulier des collocations de type Adj+Nom (easy chair, electric chair, high chair, etc.) et Nom+Nom (barber chair, beach chair, etc.), parce que ces collocations sont nécessaires pour définir la taxonomie basée sur l'hyponymie.

II.4.3. Classe des noms

Le WordNet inclut 26 catégories des noms présentées dans le tableau II.2 suivant :

Tableau II.2. Catégories des noms.

Catégorie	Exemple	Catégorie	Exemple	Catégorie	Exemple
ACT	service	GROUP	place	POSSESSION	price
ANIMAL	horse	LOCATION	area	PROCESS	process
ARTIFACT	car	MOTIVE	reason	QUANTITY	amount
ATTRIBUTE	quality	NATURAL EVENT	experience	RELATION	portion
BODY	hand	NATURAL OBJECT	river	SHAPE	square
COGNITION	way	OTHER	stuff	STATE	pain
COMMUNICATION	review	PERSON	man	SUBSTANCE	oil
FEELING	comfort	PHENOMENON	result	TIME	day
FOOD	food	PLANT	tree		

II.4.4. Relations sémantiques

WordNet est aussi structuré par des relations sémantiques entre les synsets et des relations sémantiques entre les lemmes (mots). Toutes les catégories des relations sémantiques sont utilisées à savoir ; la synonymie, l'hyperonymes, l'hyponymie, l'antonymie, la méronymie, etc.

Les méronymie est en trois types ::

X *est une composante de* Y
X *est un élément de* Y
X *est le matériau don*t Y *est constitué*

La morphologie est aussi intégrée, elle peut être de forme de flexion (exemple : woman/women, country/countries) ou de dérivation (exemple : constitution/ constitutionalize).

Le tableau II.3 suivant montre quelques exemples des relations entre les noms.

Tableau II.3. Relations sémantiques entre les noms.

Relation dans le WordNet	Relation (en français)	Appelée aussi	Définition	Exemple
Hyponym	Hyponyme	Subordonné	des concepts aux sous-types	meal → lunch
Hypernym	Hypernyme	Superordonné	des concepts aux superordonnés	breakfast → meal
Instance Hyponym	Instance-Hyponyme	Has-Instance	des concepts à leurs instances	town→ batna
Instance Hypernym	Instance-Hypernyme	Instance	des instances à leurs concepts	Hugo → author
Part Holonym	Partie-Holonyme	Part-Of	des parties aux composé	hand → body
Part Meronym	Partie-Méronyme	Has-Part	du composé aux parties	car → wheel
Antonym	Antonyme	Contraire	opposition sémantique entre lemmes	night <=>day
Derivation	Dérivation	Morpho-dérivation	lemmes de même racine morphologique	destruction<=>destroy

En pratique, le WordNet présente la relation d'hyperonymie qui relie chaque synset à ses synsets immédiatement les plus généraux et la relation d'hyponymie qui relie chaque synset à ses synsets immédiatement les plus spécifiques. Ces relations peuvent être suivies pour produire une branche d'arbre plus longues en allant au synset le plus général ou aux synsets les plus spécifiques. La Figure II.5 montre le chemin des chaînes des synsets les plus généraux pour *bass1* et *bass3* produit par la relation hyperonymie.

bass1 (the lowest part of the musical range)
=> pitch (the property of sound that varies with variation in the frequency of vibration)
=> sound property (an attribute of sound)
=> property (a basic or essential attribute shared by all members of a class)
=> attribute (an abstraction belonging to or characteristic of an entity)
=> abstraction, abstract entity
=> entity

bass3, basso (an adult male singer with the lowest voice)
=> singer, vocalist, vocalizer, vocaliser
=> musician, instrumentalist, player
=> performer, performing artist
=> entertainer
=> person, individual, someone...
=> organism, being
=> living thing, animate thing,
=> whole, unit
=> object, physical object
=> physical entity
=> entity

Figure II.5. Chaine d'hyperonymes du deux lemmes *bass1* et *bass3*

II.4.5. Classe des verbes

II.4.5.1. Définition

Les verbes sont classés par champ sémantique en trois grandes classes : état, action, évènement, où chaque classe est subdivisée en plusieurs sous classes, exemple la classe *action* est subdivisée en sous-classes de verbes de création, de changement, de communication, de compétition, de consommation … etc.

Plusieurs hiérarchies indépendantes peuvent être regroupées dans un même champ sémantique (synset), comme les verbes de possession *{have/ hold}, {receive / take} et {transfer / give}* qui sont rassemblés au sein de 3 hiérarchies différentes.

Les sens métaphoriques sont aussi inclus dans les synsets, exemple, {fail, go bad, give way, die, give out, conk out, go, break, break down}.

Des fois des sysnsets sont très proches de sens, exemple :

S1: {end, terminate} (bring to an end or halt - *mettre fin ou arrêt* -) **"She ended their friendship when she found out that he had once been convicted of a crime"** ***(****Elle a mis fin à leur amitié lorsqu'elle a découvert qu'il avait déjà été reconnu coupable d'un crime****).***

S2: {end, terminate} (be the end of; be the last or concluding part of - *être la fin de; être la dernière ou la conclusion de* -) **"This sad scene ended the movie"** ***(****Cette scène triste a mis fin au film****).***

II.4.5.2. Relations sémantiques entre les verbes

II.4.5.2.1. Hyperonymie (Hypernymie)

C'est une relation entre un verbe de l'événement à un verbe de l'événement supérieur, par exemple : *fly/ travel*

II.4.5.2.2. Troponymie

C'est une relation entre un verbe de l'événement à un verbe de l'événement subordonné, qui peut être une relation de modification par la vitesse *walk/run (marcher/courir*), par la cause *slide/pull* (glisser/tirer), par l'intensité de l'action *drowse/sleep* (somnoler/dormir) ou par la manière *move/amble* (bouger/ambler), etc.

La relation sémantique de troponymie est souvent spécifique à un champ sémantique donné, par exemple : *communicate / fax, email, phone, telex* (communicate : communiquer par un certain media), *fight / battle, war, tourney, duel, feud* (fight : se battre à une occasion particulière).

II.4.5.2.3. Implication (Entails)

l'implication est quand un verbe *v1 implique un verbe v2*, exemple *snore/sleep (ronfler/dormir)*, La relation d'implication permet aussi de structurer les verbes en une hiérarchie. Des fois l'implication est cyclique d'où elle génère une synonymie quand v1 implique un verbe v2 et v2 implique un verbe v1, exemple shut/close (fermer).

La relation peut être de cause qui relie un verbe causatif, exemple ; *to give* (donner) au verbe résultatif *to have* (avoir).

II.4.5.2.4. Antonymie

Ils existent plusieurs types d'opposition :

- L'antonymie dans le champ sémantique, exemple : *give/take, sell/buy* (donner/prendre, vendre/acheter), même action opposée.
- L'antonymie pour les verbes de changement par exemple : *lengthen/shorten* (allonger/raccourcir) et les verbes d'état, exemple : *include/exclude* (inclure/exclure).
- Les verbes opposés partagent parfois le même troponyme (avec le même super-verbe), exemple : *go up/go down, run/walk* (monter/descendre, courir/marcher).
- Certains verbes opposés partagent une implication, exemple : *fail/succeed* (échouer/réussir) implique *to try* (essayer).

II.4.6. Classe des adjectifs

Le WordNet divise les adjectifs en deux grandes catégories : les adjectifs descriptifs et les adjectifs relationnels.

- Les adjectifs descriptifs forment la plus grande catégorie.
- Les adjectifs relationnels (c'est-à-dire, ils sont reliés ou générés par dérivation à des noms, exemple : electrical), les adjectifs relationnels forment la petite catégorie.

II.4.6.1. Adjectifs descriptifs

Un adjectif descriptif assigne une valeur à un attribut d'un nom, exemple, le nom *paquet* a pour attribut *poids* dont la valeur peut être spécifiée par l'adjectif *lourd.* Le WordNet associe aux adjectifs descriptifs des pointeurs vers les noms dont ils spécifient la valeur des attributs.

La relation sémantique d'antonymie est aussi utilisée pour structurer la classe des adjectifs descriptifs.

Antonymie entre les adjectifs : L'antonymie entre les adjectifs est une relation entre mots (pas entre les synsets – sens-) à distinguer de l'opposition sémantique qui est une relation entre sens, exemple : *heavy* (lourd) est l'antonyme de *light* (léger). Ainsi de nombreux antonymes sont formés par des règles de dérivation morphologique qui s'appliquent aux mots et non aux sens, exemple : *tidy/untidy* (ordonné/ désordonné).

La similarité entre adjectifs reflète une relation de spécialisation : deux adjectifs A1 et A2 sont similaires si la classe des noms modifiée par A1 est un sous ensemble de la classe des noms modifiée par A2, exemple : *ponderous/heavy* (lourd).

Cas des quantificateurs: les quantificateurs : *few, many, some, all, less*… (peu, beaucoup, certains, tous, moins) sont traités comme des adjectifs, ils sont utilisés pour quantifier un terme, ils ont des antonymes *little/much* (peu/beaucoup) et ils sont graduables *very little, very much, less little* (très peu, beaucoup, moins peu).

II.4.6.2. Adjectifs relationnels

Ils sont dérivés morphologiquement ou reliés sémantiquement à des noms, par exemple, *electrical instrument* identifie un type d'instrument. Ces adjectifs ne sont pas graduables (very *electrical), ils* ne réfèrent pas à un attribut du nom et souvent ils n'ont pas d'antonymes.

Remarque : Certains adjectifs peuvent être à la fois descriptifs et relationnels, exemple : *innocent, innocent behavior* (innocent/comportement innocent).

II.4.7. Classe des adverbes

Les adverbes sont souvent dérivés à partir des adjectifs par l'ajout des suffixes *slow**ly**, south**ward*** (lentement, vers le sud), pareillement par l'antonymie et la gradualité exemple : *specific(ally)/ general(ly)* (spécifique(ment) /générale(ment)). Un adverbe peut appartenir à des synsets des noms ou des synsets des verbes. Les synsets des adverbes sont aussi structurés par l'antonymie.

II.4.8. Représentation graphique

La Figure II.6 montre un exemple de la représentation graphique des lemmes reliés par plusieurs relations sémantiques (Navigli, 2016).

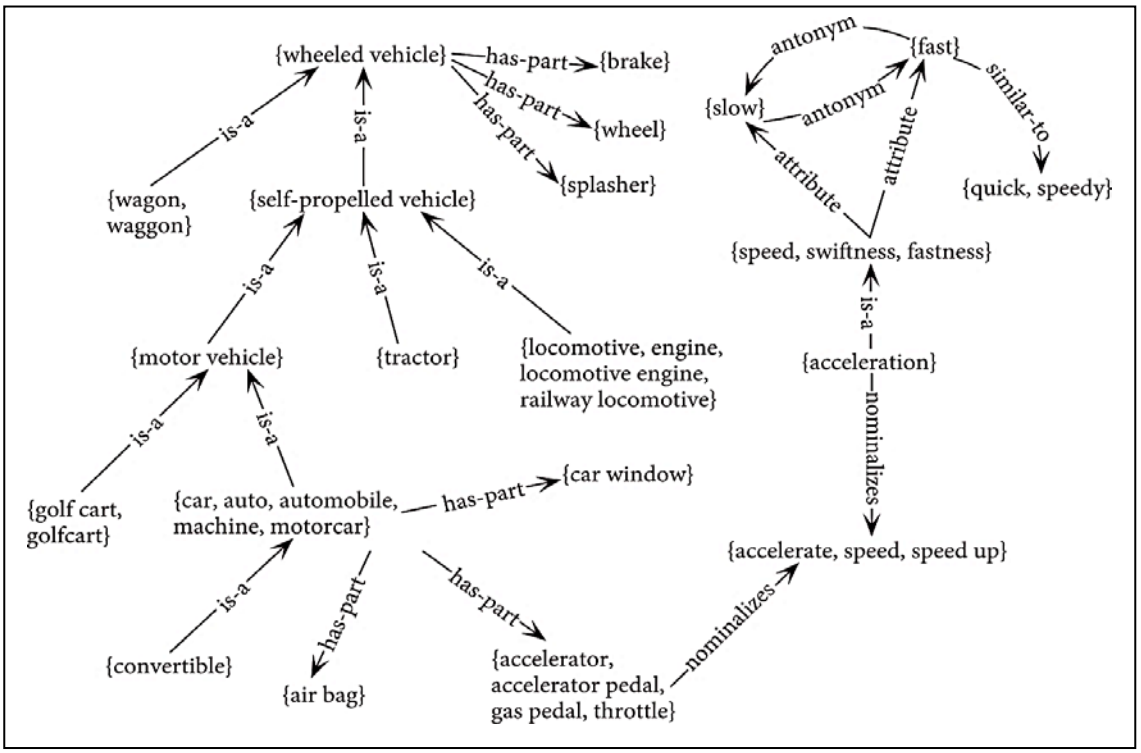

Figure II.6. Exemple de représentation graphique de WordNet (Navigli, 2016).

II.4.9. Applications du WordNet

Plusieurs méthodes ont été développées pour exploiter l'information sémantique et la nature graphique du WordNet afin de renforcer l'évolution des applications du TAL. Dès les premières applications, la similarité des mots pouvait être désignée en calculant simplement combien d'arêtes doivent être suivies pour connecter les deux mots (Leacock & Chodorow, 1998; Wu & Palmer, 1994). Pour la similarité de sysnsets (Resnik, 1995), l'auteur propose uniquement de compter le nombre maximum des mots commun entre les deux sysnsets. puis d'autres méthodes plus développées ont été construites sur le même principe (Lin & Sandkuhl, 2008). De plus, actuellement, le WordNet est toujours la ressource la plus utilisée pour la désambiguïsation des sens des mots WSD (la tâche de décider quel sens d'un mot est utilisé dans un contexte donné), et les WordNets sont toujours la base de

la plupart des évaluations dans ce domaine (Navigli, Jurgens, & Vannella, 2013). Même avec les récents développements dans le domaine du TAL, liés à l'utilisation des réseaux de neurones et d'autres méthodes, il y a eu un intérêt à exploiter la structure graphique des réseaux de mots pour développer des réseaux de neurones (Kutuzov, Dorgham, Oliynyk, Biemann, & Panchenko, 2018) et des plongements de mot (Word-Embedding) (Rothe & Schütze, 2015). D'autres méthodes et applications qui utilisent la sémantique pour résoudre la problématique de la RI ont été aussi développées et testées, ces techniques vont être présentées dans les sections suivantes.

II.5. Sémantique par vecteurs

La sémantique des données textuelles basée sur les vecteurs est fondée principalement sur le principe de la distribution initiée par la théorie du linguiste Harris (Harris, 1954), qui avait remarqué que les mots qui sont synonymes avaient tendance à apparaître dans le même contexte (avoir les mêmes mots qui apparaissent à proximité) « avec la quantité de différence de sens entre deux mots correspondant à peu près à la quantité de différence dans leur contexte » "*with the amount of meaning difference between two words corresponding roughly to the amount of difference in their environments*" (Harris, 1954).

La sémantique vectorielle est alors instanciée de ce principe à une hypothèse du plongement (embedding) de la sémantique linguistique par une représentation des sens des mots crée par les méthodes d'apprentissage automatique et d'apprentissage profond. Cette représentation sémantique vectorielle des mots est appelée plongement du mot « Word-Embedding », elle est largement utilisée dans les applications récentes du TAL qui utilisent la sémantique.

La première approche basée sur ce principe, appelée auto-surveillance (self-supervision), elle est proposée dans la tâche de modélisation du langage neuronal par (Bengio, Ducharme, Vincent, & Janvin, 2003; Collobert et al., 2011), d'où, ils ont montré qu'un réseau de neurones entrainé sur une base textuelle pouvait prédire le mot suivant uniquement sur la base de la représentation *embedding* des mots précédents. Ce modèle n'est pas conçu à une tâche spécifique de TAL, mais il servait de modèle de représentation sémantique du mot par un vecteur qui est appelé Word-Embedding.

Plus précisément, par exemple, si le mot *M1* est représenté par un vecteur V_{M1} ayant des valeurs numériques x_i, $V_{M1} = [x_1, x_2, x_3, \ldots x_{n-1}, x_n]$, et si nous voulons chercher le synonyme du mot *M1*, alors, il suffit juste de trouver un vecteur V_{M2} qui représente le mot *M2*, dont leurs vecteurs de représentation (Word-Embedding) sont similaires ou ils ont une distance de similarité minimale qui peut être déterminée par une simple formule mathématique de cosinus.

Cette idée est généralisée pour définir la sémantique vectorielle en utilisant les points pour représenter des mots dans un espace sémantique multidimensionnel. Cet espace est organisé alors par des distributions de plongement des mots (Word-Embedding), d'où les mots similaires se trouvent à des emplacements similaires, comme c'est montré par l'exemple de la Figure II.7.

Figure II.7. Exemple d'espace en 2D de *Word-Embedding* (J. Li, Chen, Hovy, & Jurafsky, 2015).

Par définition, les méthodes d'Embedding est la conversion des données textuelles d'entrée d'un réseau de neurones (c'est-à-dire les mots, les phrases, les paragraphes, les documents, les dates, les emojis, les graphiques, etc.) en nombres réels capturant la relation sémantique cachée entre ces données. Dans la littérature le Word-Embedding est aussi appelé modèle sémantique distribué, espace vectoriel sémantique distribué ou modèle d'espace vectoriel. Dans les travaux récents, plusieurs modèles et approches de Word-Embedding ont été proposés tels que ; de (Google) Word2vec, de (Stanford) GloVe, de (Facebook) FastText, et Flair…etc.

II.5.1. Word2vec

Word2Vec est un modèle de Word-Embedding créé par un algorithme d'apprentissage automatique non supervisée pour détecter les relations sémantiques et syntaxiques des mots. Le word2vec a été proposé par (Mikolov, Chen, Corrado, & Dean, 2013) chez Google, son principe est d'entraîner un simple réseau de neurones avec une seule couche cachée qui prend en entrée un grand corpus textuel brute et puis il retourne les poids de cette couche cachée sous forme d'un vecteur à des valeurs numériques. La dimension de ce vecteur est variée entre 50 à 1000. Comme c'est montré par la Figure II.8.

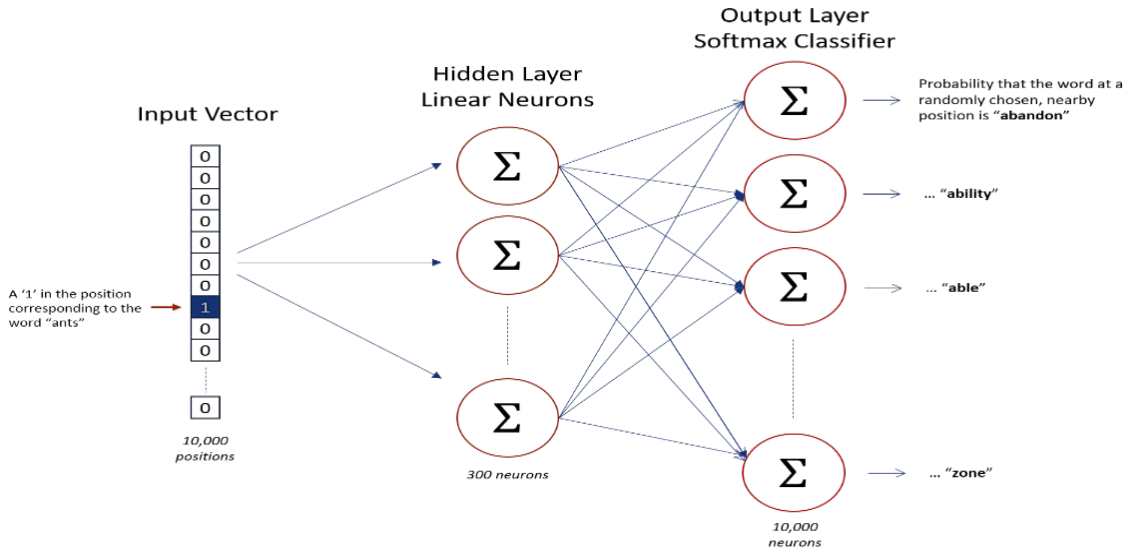

Figure II.8. Réseau de neurones pour Word2vec[7]

[7] https://petamind.com/word2vec-with-gensim-a-simple-word-embedding-example/

Le Word-Embedding de Word2vec peut être obtenu en utilisant deux modèles (tous les deux utilisent les réseaux de neurones) ; le modèle CBOW et le modèle Skip-Gram.

II.5.1.1. Modèle de sac de mots continu (CBOW)

Le Modèle de sac de mots continu CBOW (***C**ontinuous **B**ag **o**f **W**ords*) prend le contexte de chaque mot en entrée pour prédire un mot cible, c'est à dire nous fournissons au réseau de neurones des mots du contexte (les mots à gauche et à droite du mot cible) et en lui faisant prédire le mot manquant en fonction du contexte. Exemple ; si les mots : W(t-2), W(t-1), W(t+1) et W(t+2) forment un contexte (une fenêtre de deux mots à gauche et de deux mots à droite) pour le mot cible W(t) alors nous allons entrer dans le réseau de neurones le contexte W(t-2), W(t-1), W(t+1) et W(t+2) pour prédire le mot cible W(t) comme c'est illustré dans la Figure II.9.

II.5.1.2. Modèle de Skip-Gram

Le modèle de Skip-gram est l'inverse du modèle CBOW. Le Skip-Gram prend un mot pour prédire un contexte cible, l'entraînement du modèle du réseau de neurones se fait en fournissant au modèle le mot W(t) en entrée et nous demandons au modèle de prédire l'environnant W(t-2), W(t-1), W(t+1) et W(t+2) (l'ensemble des mots à gauche et à droite autour du mot W(t)) en sortie (la Figure II.9).

Dans la pratique, plusieurs chercheurs ont suggéré d'utiliser le Skip-gram pour les corpus de moins de taille par rapport au modèle CBOW qui donne une meilleure précision mais il nécessite un corpus d'entraînement plus volumineux.

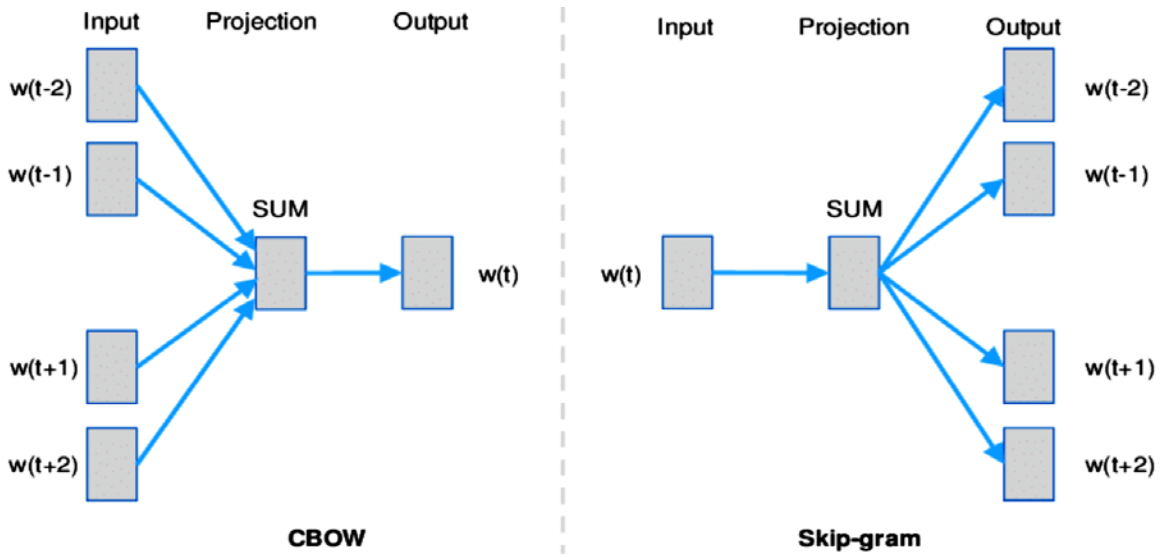

Figure II.9 Principes des modèles (CBOW) et (Skip-gram) (Mikolov et al., 2013)

II.5.2. GloVe

Les vecteurs globaux pour la représentation de mots GloVe (***Glo**bal **V**ectors for Word Representation*) (Pennington, Socher, & Manning, 2014), est un modèle d'algorithme d'apprentissage non supervisé développé à Stanford en tant que projet open source, le modèle génère des représentations vectorielles pour les mots en utilisant à la fois une factorisation matricielle globale et une fenêtre contextuelle locale. Plus précisément, GloVe crée des « *Word-Embedding* » en agrégeant des matrices globales de cooccurrence mot-mot à partir d'un corpus. Chaque élément de la matrice de cooccurrence de mots représente la fréquence dans lequel le mot *i* apparaît dans le contexte du mot *j*. Le corpus est analysé de la manière suivante, pour chaque mot cible, il recherche des mots de contexte dans une fenêtre définie par une taille de nombre de mots avant et de nombre de mots après le mot cible en accordant moins de poids aux mots les plus éloignés.

II.5.3. FastText

Le FastText est aussi un modèle de « *Word-Embedding* » créé par (Bojanowski, Grave, Joulin, & Mikolov, 2017) dans les laboratoires de recherche de Facebook, c'est une extension de word2vec qui représente les mots plus efficacement en utilisant des informations au niveau des caractères où chaque mot dans FastText est représenté lui-même par un sac de n-grammes constitutifs.

Exemple, avec n = 3 le mot <table> serait représenté par la séquence < tab, abl , ble >, puis un embedding de Skip-gram est entrainé pour chaque n-gramme constitutif et le mot *table* sera donc représenté par la somme de tous les embeddings de ses n-grammes constitutifs. L'avantage de cette technique est qu'elle traite des mots inconnus et les mots rares dans les langues à morphologie riche.

Le FastText est plus performant que Word2vec et GloVe étant donné qu'ils considèrent chaque mot comme une seule unité et ignorent la structure morphologique du mot, ce qui les rend incapables de générer de « *Word-Embedding* » pour les mots invisibles ou le vocabulaire inconnu lors de l'entraînement du modèle. FastText surmonte cette limitation puisqu'il considère chaque mot comme un n-gramme de caractères (Bojanowski et al., 2017). Autre avantage, le FastText semble être plus efficace pour le contenu des médias sociaux qui contiennent des fautes d'orthographe et des fautes de frappe (Modha & Majumder, 2019).

Une bibliothèque[8] open source FastText est disponible sur le web, comprenant des « *Word-Embeddings* » pré-entraînés pour plus de 157 langues.

[8] https://fasttext.cc

II.5.4. Flair

Flair est un modèle de « *Word-Embedding* » puissant qui capture des informations syntaxiques et sémantiques latentes qui vont au-delà des « *Word-Embeddings* » standards. Les principales différences sont les suivantes : 1) le modèle est entraîné sans aucune notion explicite des mots et il modélise fondamentalement les mots comme des séquences de caractères, et 2) il est contextualisé par le texte qui les entoure, ce qui signifie que le même mot aura des *Embeddings* différents selon son utilisation contextuelle.

La bibliothèque Flair a été développée par *Zalando Research*[9] et l'université Humboldt de Berlin[10] sous un Framework open source par PyTorch[11] pour résoudre les problématiques du TAL à savoir ; la reconnaissance des entités nommées, l'étiquetage Pos-tagging, l'analyse des sentiments et la classification des textes.

II.5.5. ELMo

ELMo (***E**mbeddings from **L**anguage **Mo**dels*) est un modèle de « *Word-Embedding* » gère le sens contextuel des mots au contraire des modelés basiques de word2vec et GloVe qui génèrent des représentations des mots séparés. Le modèle ELMo a été proposé par (Peters et al., 2018) afin de créer une représentation vectorielle intégrant tous les contextes du mot cible pour ces différentes utilisations dans les phrases, La représentation ELMo peut alors modéliser les caractéristiques syntaxiques et sémantiques du mot à sens multiples en fonction du contexte (modélisation polysémique).

Les vecteurs de mots obtenus à partir du modèle ELMo sont entrainées par la technique de l'apprentissage profond (Deep Leaning) appelée le modèle de langage bidirectionnel (*bi-directional language model*), de plus, comme les vecteurs ELMo sont basés sur des caractères, le modèle ELMo peut représenter des mots hors vocabulaire invisibles en phase d'apprentissage en utilisant des indices morphologiques (Pathak, Agarwal, Pandey, & Rautaray, 2020).

II.5.6. GLoMo

GLoMo (**G**raphs from **Lo**w-level unit **Mo**deling) est un Framework basé sur l'apprentissage non supervisé des graphes latents (Yang et al., 2018), ce modèle est aussi développé par les techniques de l'apprentissage profond (Deep Leaning) ou plus exactement les modèles de l'apprentissage par transfert (Transfer Learning) pour améliorer les performances des tâches du TAL tels que ; l'analyse des sentiments, les systèmes de question/réponse et les problèmes de la classification.

[9] https://research.zalando.com/
[10] https://www.informatik.hu-berlin.de/en/forschung-en/gebiete/ml-en/
[11] https://pytorch.org/

II.5.7. ULMFiT

ULMFiT (***U**niversal **L**anguage **M**odel **Fi**ne-**T**uning*) (Howard & Ruder, 2018) est aussi un modèle d'apprentissage par transfert (Transfer Learning) basé sur l'architecture neuronale de Bi-LSTM (*Bi-LSTM est également un modèle de l'apprentissage profond pour la classification des séquences*), il est utilisé pour les tâches de modélisation et le traitement du langage naturel. L'Embedding de ce modèle utilise des fragments des phrases et des segmenteurs de mots ce qui lui permet l'avantage de gérer les mots cachés et mal orthographiés. Son fonctionnement est indépendant par rapport à la taille des documents, les étiquettes (labels) des documents et la taille du corpus. En pratique, le modèle est pré-entraîné sur le domaine général puis affiné sur le domaine cible, il est souvent utilisé pour l'analyse des sentiments et la sémantique des textes.

II.5.8. BERT

BERT (***B**idirectional **E**ncoder **R**epresentations from **T**ransformers*) représentations d'encodeur bidirectionnel à partir de transformateurs, c'est un modèle pré-entraîné qui gère conjointement les contextes gauches et les contextes droites, crée par (Devlin et al., 2018) du groupe Google en 2018, ce modèle peut être affiné (fine-tuned) pour améliorer n'importe quelle tâche liée au TAL y compris la sémantique et les systèmes de RI. En réalité, le modèle BERT a été pré-entraîné à partir des données textuelles non étiquetées extraites d'un corpus des livres avec 800 millions de mots et de Wikipédia anglais avec 2500 millions de mots (Annamoradnejad & Zoghi, 2020). Et depuis 2019, le moteur de recherche de Google utilise BERT pour mieux comprendre les besoins des utilisateurs.

D'autres versions de BERT ont été développées relativement au TAL pour les autres langues, hors l'anglais, à savoir le mBERT (version multilingue de BERT pour 104 langues), le modèle *RoBERTa* (Y. Liu et al., 2019) de Facebook et le modèle *XLM-RoBERTa* (Conneau et al., 2019) basé aussi sur *RoBERTa* est entrainé sur 2,5 terra octets de données multilingues. Egalement, AraBert (Antoun, Baly, & Hajj, 2020) est une version qui a été réalisée pour la langue arabe.

II.5.9. OpenAI GPT

Le modèle OpenAI GPT (***G**enerative **P**re-trained **T**ransformer*) est un modèle de langage autorégressif qui utilise l'apprentissage profond pour produire un texte de type humain, créé par (Radford, Narasimhan, Salimans, & Sutskever, 2018) chez OpenAI[12], un laboratoire de recherche en intelligence artificielle basé à San Francisco. La version trois complète de GPT (GPT-3) a été introduite en mai 2020, elle a une capacité de 175 milliards de paramètres d'apprentissage, elle a été entraînée sur un très grand corpus textuel.

[12] https://openai.com/

Le GPT-3 est proposé pour la compréhension du langage naturel comprend un large éventail de tâches diverses telles que les relations textuelles, les systèmes de question/réponse, l'évaluation de la similarité sémantique et la classification des documents.

II.6. Recherche d'information sémantique

La recherche d'information sémantique se focalise sur l'utilisation des techniques sémantiques et conceptuelles afin d'améliorer la RI, ces techniques peuvent être intégrées aux différentes étapes du processus du système de RI, tels que, l'indexation, la reformulation des requêtes ou les modèles d'appariement.

II.6.1. Indexation sémantique

L'indexation dans les systèmes de RI classiques se basaient essentiellement sur le calcul de correspondance entre les mots et les documents, en revanche ce principe est vite confronté à la problématique du sens, puisqu'un sens peut être représenté par plusieurs mots et un mot peut signifier plusieurs sens, cette problématique du sens engendre donc le problème de l'ambiguïté et de la divergence des mots. Par exemple, lorsqu'un utilisateur souhaite chercher les documents relatifs aux *ordinateurs*, il serait aussi intéressé par les documents décrivant les *serveurs informatiques*, puisque ces deux notions sont sémantiquement liées par la relation de synonymie. De plus, lors de la recherche du mot général *animal*, il serait éventuellement intéressant de rechercher aussi les documents contenants le mot particulier *cheval*, puisque le cheval est sémantiquement lié à l'animal par la relation d'hypéronymie (l'hypernymie *Is-a, est-un*).

En pratique, l'indexation sémantique tente alors d'inclure tous les mots et les termes supplémentaires retrouvés par des relations sémantiques dans le descripteur de l'index du SRI. Dans la plupart des travaux de recherche sur l'indexation sémantique, ils se sont basés sur les ressources sémantiques comme les thésaurus, les ontologies ou les corpus annotés.

Au début, Voorhees (Voorhees, 1993) a proposé une technique basée sur le WordNet. Afin d'éliminer l'ambiguïté des mots, l'ensemble des synonymes du mot cible sont triés selon la valeur de cooccurrence calculée entre le contexte du mot et les mots voisins retrouvés dans les synsets de WordNet.

Une autre méthode de (Uzuner, Katz, & Yuret, 1999) qui s'est basée sur la notion de contexte pour lever l'ambiguïté des mots. Le sens d'un mot est déterminé à partir de son contexte local. La méthode s'appuyait sur le principe du contexte de Harris (Harris, 1954), elle suppose que les mots (appelés sélecteurs) utilisés dans le même contexte local ont généralement des significations similaires. Le mot sélecteur est donc utilisé pour sélectionner le synset approprié de WordNet en regardant les mots du contexte qui font partie de ce sysnset.

Des méthodes telles que l'indexation sémantique latente (LSI) (Deerwester, Dumais, Furnas, Landauer, & Harshman, 1990) peuvent projeter les termes sur un espace conceptuel

de dimensions réduites, partagé par les documents et les requêtes, qui a été appliqué avec succès dans le SRI du (Hofmann, 1999), mais au-delà d'une certaine taille, le calcul LSI devient informatiquement très couteux, car il nécessite de gérer une matrice terme-document à grande échelle. Plus tard, des méthodes de réduction de dimensionnalité prises des approches des modèles thématiques ont été appliquées par (Yi & Allan, 2009) telles que l'allocation de Dirichlet latente (Wei & Croft, 2006) et le modèle d'allocation appelé Pachinko (W. Li & McCallum, 2006).

Les travaux de (Priss, 2000) utilisaient des treillis de concepts pour améliorer la représentation d'une collection de documents, en la fusionnant avec des informations provenant de thésaurus et créant ainsi un contexte étendu à multiples facettes. Dans une approche similaire, les travaux de (Carpineto, Romano, & Bordoni, 2004) ont présenté un système qui interroge le moteur de recherche de Google pour construire un navigateur à facettes à partir d'un treillis de concepts pour aider l'utilisateur à travers de son expérience de recherche.

(Khan, McLeod, & Hovy, 2004) a proposé une autre méthode d'indexation appliquée sur des concepts d'ontologie du domaine de sport. L'approche commence par l'identification des mots descripteurs à partir du texte, puis une comparaison de ces mots avec les concepts de l'ontologie afin de déterminer les mots appropriés pour enrichir l'index. L'approche propose une désambiguïsation des concepts en se basant sur la distance sémantique en calculant le score entre le concept ambigu et les autres mots du contexte. Le concept qui obtient le score de distance minimum est retenu.

Une autre approche automatisée, elle utilisait des ontologies de documents comme source de représentation conceptuelle. (Gauch, Madrid, Induri, Ravindran, & Chadlavada, 2004) utilisent KeyConcept, c'est un système de RI qui schématise les documents à un sous-ensemble limité des concepts, en utilisant la catégorisation des documents selon ces concepts en tant que données d'entraînement pour les classificateurs de concepts, et en effectuant une recherche sur le texte augmenté par la représentation conceptuelle.

(Baziz, Boughanem, & Aussenac-Gilles, 2005) a proposé le modèle DocTree qui a pour but de représenter le contenu sémantique des documents par un réseau sémantique qui est composé par les concepts (nœuds) et des relations entre eux, qui sont les arcs. Ce réseau est créé par la projection du contenu textuel du document sur le WordNet, en comparant les mots du document avec les mots des synsets, et ceux qui ont une correspondance sont retenues comme des concepts, les auteurs ont aussi proposé une technique pour la désambiguïsation de ces concepts calculée par Ctf*idf. Pareillement, Dans les approches proposées par (Köhler, Philippi, Specht, & Rüegg, 2006; Su & Gulla, 2006), le lexique de WordNet est utilisé pour lemmatiser les mots des documents pour créer les index. L'analyse formelle de concepts a été aussi appliquée à l'indexation de documents dans (Manning, Prabhakar, & Hinrich, 2009) pour exploiter les relations que les documents (objets) entretiennent à travers les termes qu'ils partagent (attributs).

Le travail de (Hernandez, Hubert, Mothe, & Ralalason, 2008) propose une combinaison entre l'indexation basée sur des mots-clés du document et l'indexation basée

sur des termes identifiés depuis des synsets du WordNet. Les résultats ont démontré une amélioration de 16% dans le rappel et de 4 % dans la précision.

Le projet Menelas (Hernandez et al., 2008) a développé un système d'accès aux rapports médicaux des hôpitaux. Il s'articule autour d'une ontologie construite à partir de rapports à indexer incluant l'ensemble des maladies coronariennes.

(Harrathi, Roussey, Maisonnasse, & Calabretto, 2010) proposent une étude pour l'indexation sémantique des documents multilingues, où les termes simples sont extraits par la méthode classique d'indexation, tandis que les termes composés sont identifiés par une mesure statistique basée sur la fréquence des mots qui apparaissent mutuellement dans le document. Les termes qui dépassent le seuil sont projetés sur une ontologie sémantique multilingue. L'approche de Harrathi est évaluée dans un Système de RI basé sur le modèle de langage proposé par (Maisonnasse, Gaussier, & Chevallet, 2009) utilisant la ressource médicale UMLS[13] (Unified Medical Language System) et la collection de test multilingue (anglais, allemand et français) CLEFmed2007[14]. Les résultats obtenus ont donc montré une amélioration moyenne de précision de 5 % par rapport à l'indexation classique.

(B.-D. Dinh, 2012; D. Dinh & Tamine, 2010) ont présenté une approche d'indexation sémantique pour le domaine biomédical s'appuyant sur les concepts du thésaurus MeSh[15]. L'approche commence par l'extraction de concepts d'un document et d'une requête respectivement, en projetant son contenu textuel sur une liste prédéterminée de tous les concepts appartenant au thésaurus MeSh. Un score est ensuite attribué à chacun des concepts des termes candidats en fonction de sa similarité thématique avec le texte et de sa similarité structurelle définie par le degré de corrélation entre son entrée dans le thésaurus et le contexte du terme dans le texte. Les concepts ambigus sont désambiguïsés par deux approches. Cette technique a été évaluée sur la collection OHSUMED[16] de revues médicales, dont les résultats ont montré une amélioration dans les performances de 17,35 % pour la désambiguïsation pas à pas et 17,06 % avec la désambiguïsation appuyé par la catégorisation (clustering).

L'étude de (Egozi, Markovitch, & Gabrilovich, 2011) propose une méthode qui augmente la représentation de texte basée sur des mots-clés avec des features (caractéristiques) basés sur des concepts, extraits automatiquement du Wikipedia. L'approche génère automatiquement des nouveaux features de texte par des méthodes qui utilisent des données d'apprentissage étiquetées autogénérées. Le système résultant est évalué sur plusieurs collections de TREC, montrant des performances supérieures par rapport aux résultats précédents.

(Mallak, 2011) propose d'indexer les documents et les requêtes par des clusters de concepts, basé sur le principe proposé par (Baziz et al., 2005) mais en utilisant la technique de centralité conceptuelle pour la désambiguïsation des concepts. Afin de sélectionner un document pertinent pour une requête, le modèle d'appariement compare alors le graphe associé à des clusters de documents en utilisant une mesure qui combine trois facteurs; la

[13] http://www.nlm.nih.gov/research/umls/.
[14] http://www.clefcampaign.org/.
[15] http://www.nlm.nih.gov/mesh/
[16] http://trec.nist.gov/data/t9_filtering.html.

centralité du concept au sein du cluster, sa fréquence relative dans le document et sa spécificité définie par sa profondeur dans la hiérarchie « *is-a est-un* » de WordNet.

(Boubekeur & Azzoug, 2013) ont aussi proposé l'indexation par les sens des mots, le système commence par extraire des termes descriptifs, simples ou composés à partir des documents, en projetant le texte du document sur le WordNet, pour la désambiguïsation ils se sont basé sur la notion de centralité du concept. Leur étude était sur la collection de test Time, composée de 423 documents et 83 requêtes.

Plus récent, l'impact des approches des réseaux de neurones pour l'indexation et la recherche d'information sémantique, et les systèmes de question/réponse dans le domaine de biomédicale ont également montré des résultats plus probants dans le challenge BioASQ (Kakadiaris, Paliouras, & Krithara, 2018; Nentidis et al., 2020). De plus, le travail de (Neji, Jemni Ben Ayed, Chenaina, & M Shoeb, 2021) a introduit un nouveau modèle de pondération conceptuelle pour l'indexation sémantique. La formule de pondération proposée dépendait de divers facteurs, tels que les degrés de localité et d'intégralité du concept. En outre, le travail a exploité la tautologie du concept, le degré de spécificité du concept et un paramètre correcteur pour atténuer l'incertitude résultant du processus de WSD. Le processus comprenait trois étapes principales : l'identification du concept, l'indexation et la notation des documents. Les résultats ont montré que l'approche proposée surpassait les autres modèles basés sur les techniques classiques.

II.6.2. Reformulation sémantique des requêtes

Dans la recherche d'information, L'expansion (la reformulation) des requêtes est l'une des techniques indispensables pour surmonter les problèmes d'inadéquation des mots-clés de la requête ainsi pour se rapprocher au besoin de l'utilisateur, cette technique est subdivisée principalement en trois grandes catégories en fonction des méthodes et des ressources d'expansion : Premièrement, les approches basées sur les informations issues d'un ensemble de documents initialement récupérés par le système de RI (Approches locales ou réinjection de la pertinence *feedback*). Deuxièmement, les approches basées sur des informations globales issues de tous les documents de la collection (Approches globales). Troisièmement, les approches basées sur des structures de connaissances comme des corpus, des thésaurus, des dictionnaires ou à travers leurs combinaisons. Plusieurs approches de l'expansion des requêtes ont été étudiées dans le passé. Les premiers travaux ont été menés par (Jones, 1971) qui avait utilisé des mots groupés basés sur la cooccurrence dans les documents pour élargir la requête. Ensuite, des techniques d'analyse locale et globale ont été utilisées par (Xu & Croft, 2000). (Carpineto & Romano, 2012) a présenté ainsi un Survey intéressant détaillant ces différentes approches.

II.6.2.1. Analyse globale

L'analyse globale de la collection est effectuée pour obtenir la corrélation sémantique entre tous les mots du corpus et les mots-clés originaux de la requête, ensuite les mots qui ont plus de similarités en dépassant un seuil donné sont sélectionnés comme candidats à l'expansion de la requête. Les résultats expérimentaux sur la collection TREC ont montré l'efficacité de l'approche proposée par (Hu, Deng, & Guo, 2006).

II.6.2.2. Analyse locale

L'analyse locale est effectuée uniquement dans les top documents récupérés par le système de RI pour une requête initiale, connus sous le nom de la réinjection de la pertinence, les mots candidats d'expansion sont sélectionnés en calculant les similarités à partir de l'analyse des passages de texte (une fenêtre de texte avec une taille fixe) comme c'est proposé dans (Xu & Croft, 1996). L'analyse locale peut généralement obtenir des performances plus élevées par rapport à l'analyse globale, mais les résultats de cette approche dépendent toujours par la récupération initiale du SRI (Huang, Wang, & Zhang, 2011). Des études approfondies sur les méthodes de la réinjection de la pertinence ont montré de meilleurs résultats, comme les travaux de (Colace, De Santo, Greco, & Napoletano, 2015), où les auteurs ont proposé une méthode d'expansion de requête qui extrait automatiquement un ensemble de paires de mots pondérées à partir d'un ensemble de documents liés au topic fournis par le retour de pertinence et par le calcul des probabilités. Autres évaluations réalisées ont démontré l'efficacité par rapport aux approches initiales, à savoir le travail présenté par (Karisani, Rahgozar, & Oroumchian, 2016) qui a proposé une méthode pour identifier et re-pondérer l'informativité des termes de la requête, en examinant la similarité des mots des top-documents et leurs pondérations selon leurs contextes. L'analyse des résultats obtenus a indiqué que la méthode suggérée est capable d'identifier les mots-clés les plus importants même dans les requêtes courtes et elle améliore les performances du système de RI environ 7% de MAP par rapport aux méthodes traditionnelles.

II.6.2.3. Thésaurus

Les thésaurus, en tant que ressources externes, ont été largement utilisés dans les approches d'expansion de requête. Les travaux de (Stairmand, 1997) basés sur le WordNet ont conclu que la performance était limitée par la couverture du WordNet et le processus d'expansion est étroitement lié à la richesse de cette ressource. Pour les travaux basés sur WikiPedia, les auteurs dans (Yinghao Li, Luk, Ho, & Chung, 2007) ont développé une méthode d'expansion de requêtes basée sur cette ressource en utilisant des catégories de documents, dans lesquelles le poids de la catégorie est attribué aux mots-clés initiaux de la requête. Khoury (Khoury, 2011) a utilisé la même ressource pour identifier les termes synonymes et les entités linguistiques sémantiquement similaires aux mots-clés originaux et il les ajoute à la requête avant de commencer la recherche par le SRI. Aggarwal et Buitelaar (Aggarwal & Buitelaar, 2012) ont aussi extrait des connaissances de Wikipedia et DBpedia pour l'expansion des requêtes. Le travail de (Egozi et al., 2011) a présenté une autre technique combinée, en utilisant un thésaurus construit à partir des documents récupérés par le SRI pour une première recherche (méthode locale).

II.6.2.4. Méthodes basées sur des concepts

Plusieurs travaux dans la littérature sont consacrés à l'étude de l'expansion de requêtes basées sur des concepts extraits d'ontologies. Lorsque les chercheurs utilisent des concepts plutôt que des termes isolés, ils essaient d'exprimer des sens implicites qui ne sont pas donnés par des termes isolés, un concept est donc représenté comme un ensemble de termes adjacents. Les auteurs dans (L. Liu, Cao, Zhang, & Tian, 2009) ont proposé une reconnaissance de sens des hyponymes basée sur l'espace conceptuel, dont ils ont utilisé les contextes d'hyponymes et les poids des features (caractéristiques) des mots pour construire un espace vectoriel hyponyme-mot. Aseervatham (Aseervatham, 2009) a également étudié un modèle d'espace vectoriel conceptuel (Concept Vector Space Model CVSM), qui utilise les connaissances linguistiques antérieures pour capter la signification des documents. Autre étude de Huang (Huang et al., 2011) a présenté une nouvelle approche, dont l'idée est de construire un TASM « Tree of Associational Semantics Model » et de sélectionner des mots-clés candidats dans l'arbre, les expériences montrent que le résultat de cette approche est meilleur par rapport aux méthodes traditionnelles basées sur tf*idf.

Récemment, Aklouche et ses co-auteurs ont présenté dans leurs travaux plusieurs méthodes, à savoir ; La méthode de la réinjection de pertinence aveugle basée sur des graphes de cooccurrence en mesurant la similarité terme-terme (Aklouche, Bounhas, & Slimani, 2019) et la méthode d'expansion de requêtes basées sur le modèle Word2vec de « Word-Embedding », les scores obtenus pour les résultats globaux montrent que 80% des topics sont au-dessus des scores médians du TREC (Aklouche, Bounhas, & Slimani, 2018).

II.7. Conclusion

Dans ce chapitre, nous avons détaillé la sémantique dans les textes, où nous avons présenté la définition du sens, les différentes relations sémantiques, les techniques pour la désambiguïsation des sens des mots (WSD). Nous avons aussi exposé en détail la ressource sémantique du WordNet et la représentation sémantique par les approches vectorielles et le Word-Embedding tel que ; les modèles de Word2vec, GloVe, FastText, Flair, ELMo, Bert, etc.

Ensuite, nous avons exposé les différents travaux connexes dans la littérature qui traitent la problématique de la sémantique dans le domaine de la recherche d'information en proposant des solutions pour améliorer les performances des moteurs de recherche ou les systèmes de RI pour les deux phases ; la partie de l'indexation des documents et la partie de la reformulation des requêtes. Les différentes améliorations dans les résultats obtenus par ces travaux ont démontré l'importance de la RI sémantique pour améliorer le processus général des systèmes de RI.

Chapitre III. Recherche d'information arabe: Outils et ressources

III.1. Introduction

Dans ce troisième chapitre, nous nous intéressons à présenter l'importance de la langue arabe, ses caractéristiques, ses challenges et ses différents outils, ressources et softwares informatiques disponibles liés au texte ou au langage qui aident son développement dans le domaine du traitement automatique de la langue (TAL) et la recherche d'information (RI). Ainsi une partie dans ce chapitre est consacré pour la présentation des travaux connexes réalisés dans le domaine de la recherche d'information arabe dans ses différentes phases à savoir ; La phase de reformulation de requêtes et la phase de l'indexation de la base documentaire réalisées par les techniques utilisant les principes de la morphologie du texte, des méthodes de n-grammes ou à travers les ressources externes.

III.2. Langue arabe

L'arabe est la langue sémitique parlée par plus de 300 millions de locuteurs dans vingt-trois pays à travers l'Afrique du nord et l'Asie du sud-ouest [1]. Cela fait de l'arabe la cinquième langue la plus répandue dans le monde. L'arabe est aussi la langue parmi les six langues officielles de l'ONU[2] (l'anglais, l'arabe, le chinois, l'espagnol, le français et le russe).

L'arabe a une grande influence qui est principalement due à: i) des raisons religieuses, où l'arabe est la langue du Coran et de l'érudition islamique; et ii) L'arabe était la langue de la science et de la technologie au moyen âge, les principales universités arabes d'Espagne, d'Afrique et du Moyen-Orient étant des centres de lumières scientifiques et d'apprentissage. Même actuellement, l'arabe fait partie des programmes scolaires dans la plupart des pays musulmans majoritairement non arabes comme le Pakistan et l'Iran. L'arabe est également une langue officielle dans d'autres pays tels que l'Érythrée, le Tchad et la Somalie. L'arabe a eu une influence, principalement en termes de vocabulaire, sur de nombreuses autres langues telles que le persan, le turc, l'ourdou, l'espagnol, le swahili (c'est une langue d'origine bantoue principalement métissée avec l'arabe, parlé en Afrique de l'Est.) et l'haoussa (langue africaine parlée en Afrique de l'Ouest, principalement au Nigeria et au Niger). De plus, l'écriture arabe est utilisée dans nombreuses langues telles que le persan, le kurde, le dari, le pashto et l'ourdou (Darwish & Magdy, 2014).

[1] https://iedja.org/monde-ou-pays-arabes/
[2] https://www.un.org/fr/sections/about-un/official-languages/index.html

L'arabe possède aussi de nombreuses propriétés et caractéristiques, citons :

1. **L'Arabe Standard Moderne** (en anglais : **Modern Standard Arabic -MSA-**), appelé en arabe par (العربية الفصحى الحديثة- *al-ʿarabiyya al-fuṣḥā al-ḥadīṯa),* c'est la langue officielle du monde arabe, elle est aussi la langue principale et fonctionnaire de l'éducation, de l'enseignement, des institutions officielles et des médias. Le MSA est basé syntaxiquement, morphologiquement et phonologiquement sur l'arabe classique (ancienne langue de la poésie préislamique), la langue du Coran et l'arabe classique post-coranique ("langue de la civilisation arabo-musulmane, appelée, en arabe, *arabe pur patrimonial* : العربية الفصحى التراث- *al-ʿarabiyya al-fuṣḥā al-turāṯ"*). Le MSA est caractérisé aussi par une grande flexibilité grammaticale; les mots peuvent être arrangés de différentes manières (Habash, 2010; Khatib, 1997).
2. Le langage écrit est composé de vingt-huit lettres (16 lettres d'entre elles ont un point, deux ou trois points).
3. Les formes des lettres changent en fonction de leurs positions dans le mot (au début, au milieu ou à la fin du mot). Les lettres peuvent être attachées les unes aux autres ou non (isolées), le tableau III.1 suivant montre un exemple sur ce changement de forme pour la lettre (qaf-ق).

Tableau III.1. Représentation de la lettre (qaf-ق) selon ses positions dans le mot.

Au début	Au milieu	A la fin	Isolée
قلعة Château	طقس Météo	حلق Gorge	فرق Différence

4. Le sens de l'écriture et de la lecture est de droite à gauche.
5. Les signes diacritiques (tachkil التشكيل) sont facultatifs (qui sont souvent omis à l'écriture ce qui génère plus d'ambiguïté des mots): le texte arabe écrit peut être entièrement diacritisé (par exemple, l'arabe classique, y compris les textes historiques, liturgiques et didactiques), partiellement diacritisé ou entièrement non diacritisé, notamment, les textes écrits dans les livres et les journaux ne possèdent pas habituellement de signes diacritiques. Aussi, la diacritisation se fait en dessus et en dessous des lettres selon plusieurs formes, comme c'est montré dans le tableau III.2 suivant :

Tableau III.2. Formes de diacritisation (tachkil).

Double consonne (intensité) "الشدة"	Aucune voyelle "سكون "	Nunnation " تنوين"			Courte voyelle "حركة"		
فّ /ff/	فْ /f/	فٍ /fine/	فٌ /foune/	فً /fane/	فِ /fi/	فُ /fou/	فَ /fa/

6. Les mots arabes sont généralement dérivés des racines composées de deux, trois ou quatre lettres, les racines à trois lettres (trilittérales) étant les plus courantes. La construction des mots se fait par l'insertion éventuellement des affixes, en ajoutant des préfixes, des suffixes, des infixes, ou en doublant des constantes comme montre le tableau II.3 suivant.

Tableau III.3. Dérivation des mots depuis le mot arabe *Ecrire* (كتب).

Ecrire	Ecrivain	Livre	Petit livre	Ecrit	Abonné
كتب	كاتب	كتاب	كتيب	مكتوب	مكتتب

7. La langue écrite n'a pas de majuscules pour les noms propres, comme les noms de personnes, les positions géographiques, les villes, les pays, les mois et les jours de la semaine, etc.
8. Les noms peuvent être au singulier, au pluriel ou au double, notamment au masculin ou au féminin.
9. Ces caractéristiques contribuent à générer une ambiguïté accrue et complexe, en particulier dans les tâches automatisées, telles que l'analyse du texte, l'indexation automatique, l'extraction d'informations, la reconnaissance d'entités nommées, etc.

III.3. Outils et ressources pour la recherche d'information arabe

Le développement des Systèmes de Recherche d'Information (SRIs) Arabe implique plusieurs tâches de traitement et d'analyse de texte telles que la segmentation et la tekenisation du texte brut (pour les documents et les requêtes), la spécification des étiquettes et des marques des mots (Part_Of_Speech *Pos_Tagging*) de la partie du discours, l'analyse morphologique (lemmatisation ou radicalisation), la désambiguïsation lexicale, morphologique, syntaxique et sémantique , la diacritisation, la reconnaissance des entités nommées, l'analyse syntaxique (par phrase ou par syntagme verbal ou nominal)… etc. L'implémentation de toutes ces tâches via les langages de programmation à partir de zéro nécessite un travail lourd et fastidieux, une meilleure solution est l'utilisation et l'intégration dans nos programmes principaux des bibliothèques (outils) et des ressources de TAL et de linguistique proposées par plusieurs plateformes et packages.

Nous décrivons dans ce qui suit les types d'outils et de ressources linguistiques nécessaires pour implémenter les différentes tâches et fonctions d'un SRI. Ensuite, nous présentons une panoplie d'outils et de ressources linguistiques disponibles gratuitement, qui peuvent être ainsi utilisés dans le développement des composants des SRIs pour l'arabe.

III.3.1. Outils et ressources pour le traitement et l'analyse des textes

III.3.1.1. Outils nécessaires pour traitement et analyse de texte

- **Outils de segmentation de texte**: les outils de segmentation de texte divisent un document textuel en unités significatives et analysables telles que des symboles, des mots et des phrases. Les outils de segmentation de texte les plus utilisés sont les segmenteurs (tokenizers).
- **Outils de l'étiquetage et l'annotation de texte**: (en anglais word tagging) ce sont des outils de prétraitement importants qui fournissent des informations supplémentaires afin d'améliorer la précision des systèmes de RI, parmi les outils les plus utilisés sont les étiqueteurs POS-Tag (Part_Of_Speech) du discours et les outils de reconnaissance des entités nommées.
- **Outils d'analyse de mots**: les outils d'analyse de mots transforment les mots en leurs composants linguistiques initiaux ou radicaux afin de trouver le mot exact dans son contexte (en particulier, les mots-clés). Etant l'orthographe arabe est très ambigüe, ces outils aident à la désambiguïsation lexicale et morphologique. Ces outils sont utilisés pour améliorer les indexes des collections, annoter les textes avec des caractéristiques morphologiques (features) et pour récupérer les documents dans la phase de recherche du SRI. Parmi ces outils sont les analyseurs morphologiques et les radicalisateurs (stemmers).
- **Outils d'analyse syntaxique**: les outils d'analyse syntaxique sont des programmes qui analysent la grammaire et la structure syntaxique des phrases ou des syntagmes dans le texte. Par exemple les analyseurs syntaxiques (Parsers ou Tree-tagger).
- **Outils d'analyse sémantique**: les outils d'analyse sémantique sont utilisés pour détecter la signification ou le sens des mots recherchés en fonction du contexte et des caractéristiques linguistiques (linguistic features). Ces outils peuvent inclure d'autres outils de désambiguïsation sémantique (comme les WSD) et de conceptualisation des termes.

En plus de ces outils d'analyse de texte, l'implémentation et la construction des SRIs nécessitent d'autres outils et plateformes supplémentaires, tels que:

- **Outils de recherche d'information**: ce sont les bibliothèques et les plateformes des moteurs de recherche d'information qui sont utilisées pour l'indexation, la reformulation des requêtes et le classement des documents, comme *Lucene* pour Java et *Whoosh* pour Python.
- **Outils et environnements de développement de TAL**: Les outils de développement du TAL sont des boîtes à outils (Toolkits) qui se composent de plusieurs bibliothèques et programmes destinés à aider les spécialistes du domaine

et les programmeurs à développer et analyser d'autres outils, par exemple GATE[3] et NLTK[4].

III.3.1.2. Ressources linguistiques pour la recherche d'information

- **Ressources de connaissances :** ce sont des bases de connaissance qui aident à la conceptualisation de mot ou groupe de mots qui peuvent être utilisées généralement pour la détection de la sémantique ou mesurer la similarité entre deux segments de texte, d'extraire des relations sémantiques ou de trouver des mots sémantiquement équivalents. Plusieurs ressources de connaissance sont disponibles comme les thésaurus, les ontologies, le wiki, etc.
- **Outils de détection de langue :** c'est très important de connaitre la langue utilisée pour la recherche d'information pour orienter le SRI vers les index appropriés.
- **Corpus linguistiques :** les corpus sont formés généralement par des gros textes annotés qui sont utilisés pour l'apprentissage et le test des systèmes de TAL. Exemple le Treebank[5] est un corpus hautement structuré composé de grandes collections d'analyses syntaxiques pour des phrases annotées et vérifiées manuellement.
- **Collections de test des SRIs :** ce sont des collections formées par un ensemble des requêtes, une base documentaire textuelle et des jugements de pertinences de ces requêtes par rapport à cette base documentaire.
- **Listes des mots vides :** ils sont aussi des ressources nécessaires pour le prétraitement dans la phase de l'indexation et l'analyse des requêtes.

Ces dernières années, un grand nombre d'outils ont été développés pour les tâches et les opérations de traitement de texte (TAL). Plusieurs parmi eux sont, également disponibles gratuitement pour la communauté des chercheurs et qui peuvent être ainsi utilisés pour l'implémentation et mise en œuvre des SRIs pour l'arabe.

La disponibilité d'outils gratuits pour la communauté de recherche réduira considérablement le coût du développement des SRIs pour l'arabe. Il convient également de noter que les résultats expérimentaux de la recherche peuvent être plus facilement comparés les uns aux autres lorsqu'ils s'appuient sur les mêmes ressources et/ou outils accessibles au public. Nous présentons dans les sous-sections suivantes des outils et des ressources linguistiques disponibles gratuitement. Ces outils aident à la conception, l'implémentation, le développement, les tests et l'analyse comparative des approches et des méthodes du SRI.

La présentation des sections suivantes est organisée pour répondre aux questions suivantes ; quelles sont les descriptions et les caractéristiques de l'outil ? Qui a conçu l'outil (ou la ressource) ou le travail de recherche qui l'a développé ? Quelle technique a été utilisée pour concevoir l'outil? Où pouvons-nous télécharger cet outil (s'il est disponible)?

[3] https://gate.ac.uk/
[4] https://www.nltk.org/
[5] https://copticscriptorium.org/treebank.html

III.3.2. Outils d'analyse des textes arabes

Comme c'est présenté préalablement, dans l'étape d'analyse des requêtes, le SRI prend la requête en entrée, supprime les mots vides, extrait les mots-clés et reformule enfin cette requête, par exemple en ajoutant des mots sémantiquement équivalents. Pour la phase d'indexation des documents, le SRI procède par la tokenisation (segmentation), la suppression des mots vides, l'étiquetage POS_tag, la radicalisation (stemming), la normalisation. Ainsi, le SRI a besoin des analyseurs morphologiques et des ressources pour trouver des mots sémantiquement équivalents pour enfin concevoir le modèle d'appariement approprié *requête-index*. Nous présentons dans ce qui suit les différents outils et ressources liés à toutes ces taches.

III.3.2.1. Stanford Word Segmenter

Stanford Word Segmenter[6] est un outil autonome développé à l'université de Stanford, USA en utilisant la plate-forme Java, pour la tokenisation du texte brut écrit en chinois et en arabe. Cet outil traite le texte brut arabe selon la norme *Penn Arabic Treebank (PATB)* (Maamouri, Bies, Buckwalter, & Mekki, 2004) version 3 et il produit comme sortie une liste des tokens (mots). Cet outil est également disponible pour une utilisation dans .NET Framework. Le *Stanford Word Segmenter* nécessite que le fichier texte arabe, qui doit être segmenté, soit encodé en UTF-8. Les programmeurs ont aussi la possibilité de modifier le code source Java de *Stanford Word Segmenter* pour toutes les exigences spécifiques ou de l'intégrer en tant que composant dans d'un SRI, soit dans la partie requête ou dans la partie d'indexation de la base documentaire.

III.3.2.2. Stanford Log-linear POS Tagger

Stanford log-linear POS tagger[7] est un autre outil développé à l'université de Stanford, USA en utilisant la plate-forme Java pour lire du texte brut et attribuer une étiquette POS-Tag à chaque mot (token). La version complète de cet outil est liée à l'étiqueteur d'anglais (*English tagger*), dont elle fournit trois modèles d'étiqueteurs, un modèle d'étiqueteurs chinois, un modèle d'étiqueteurs en allemand et un modèle d'étiqueteurs en arabe. Bien que cet outil peut être aussi adapté à d'autres langues. Le modèle arabe de cet étiqueteur a été entraîné aussi sur l'ensemble des standards *PATB* version 3. L'étiqueteur *POS log-linéaire de Stanford* a été utilisé pour les tâches de reformulation des requêtes et de l'indexation des documents.

III.3.2.3. Mots vides arabes (Stopwords)

La suppression des mots vides (stoplist ou stopword) est une technique qui consiste à éliminer les mots non utiles et trop fréquents dans la phase d'indexation de la base documentaire pour réduire la taille de l'index et aussi dans la phase d'analyse de la requête. Ces mots sont regroupés dans une liste appelée en anglais stoplist ou la liste des mots vides.

[6] https://nlp.stanford.edu/software/segmenter.html
[7] https://nlp.stanford.edu/software/tagger.html

Les mots vides arabes sont disponibles en grand nombre, en raison de la richesse du lexique de la langue par son grand nombre de mots et de leurs dérivés. La liste des mots vides inclue certains liens grammaticaux tels que l'article défini (le - ال), les prépositions attachées et séparées, les conjonctions, les mots interrogatifs, les mots négatifs, les exclamations, les lettres d'appel, les adverbes de temps et de lieu. Ils comprennent également tous les pronoms, les démonstratifs, les pronoms sujet/objet, les cinq noms distinctifs, certains nombres, les ajouts et certains verbes.

Malgré que la suppression des mots vides est une source d'ambigüité, plusieurs travaux ont montré qu'elle aide à augmenter le rappel du SRI pour l'arabe. Plusieurs listes des mots vides arabes sont disponibles gratuitement soit sur les librairies comme NLTK, intégrées dans des moteurs de recherche comme *Lemur* et *Solr* ou via le web, entre autre *Al-mostabaadat (المستبعدات)*[8], *Anton Balucha stopword list*[9], *Arabic stopword list from UniNE*[10], *Wael Salloum stopword list*[11].

III.3.2.4. Light stemmer

Light stemmer ou l'algorithme de la lemmatisation légère est un algorithme très utile dans le prétraitement du texte, cet outil a été utilisé par plusieurs travaux liés à la RI. Dans lequel le programme de *Light stemmer* supprime aveuglement tout simplement les affixes, les préfixes (les ajouts au début du mot), les infixes (les ajouts au milieu du mot) et les suffixes (les ajouts à la fin des mots), Exemple : طفل, اطفال, طفلهم (enfant, enfants, **leur** enfant). L'inconvénient de cet algorithme est que parfois il génère des ambiguïtés, par exemple le mot متطفل (parasitaire) génère le mot ambiguë طفل (enfant). Plusieurs bibliothèques pour cet algorithme sont disponibles et téléchargeables gratuitement sut le web[12], où elles sont directement intégrées sur les moteur de recherche Lemur[13] ou Solr[14].

III.3.2.5. Khoja Stemmer

La lemmatisation (stemming) est une tâche importante car certains SRIs incluent les mot-clés radicaux et les mots en formes fléchies (dérivationnelles et flexionnelles) dans leurs index afin d'améliorer le rappel dans la recherche et certains d'autres n'incluent que des mot-clés radicaux (des lemmes).

Le lemmatiseur de *Khoja stemmer* a été utilisé dans le cadre d'un SRI créé et implémenté à l'université du Massachusetts, USA, pour la piste multilingue TREC-10 en 2001 (Larkey & Connell, 2001). Il se base sur la lemmatisation légère, dans lequel il supprime les suffixes et les préfixes les plus long et puis il fait correspondre le mot restant avec des modèles verbaux et nominaux pour extraire la racine. Ce lemmatiseur utilise

[8] http://sourceforge.net/projects/arabicstopwords/
[9] https://code.google.com/p/stop-words/
[10] http://members.unine.ch/jacques.savoy/clef/index.html
[11] https://www.academia.edu/2663620/A_Modern_Standard_Arabic_Closed-Class_Word_List
[12] https://arabicstemmer.com/
[13] https://sourceforge.net/p/lemur/wiki/Parser%20Applications/
[14] https://cwiki.apache.org/confluence/display/solr/LanguageAnalysis

plusieurs fichiers de données linguistiques tels que la liste des lettres arabes, la liste des caractères de ponctuation et une liste de 168 mots vides. Le *Khoja stemmer* traite aussi l'article défini (le ال) "al atta3rif". Une implémentation Java de son algorithme de lemmatisation est disponible sur le Web[15].

III.3.2.6. Information Science Research Institute's (ISRI) Stemmer

Le *ISRI Arabic Stemmer* "The **I**nformation **S**cience **R**esearch **I**nstitute" (ISRI) Arabic stemmer est un lemmatiseur pour la langue arabe. L'algorithme de ce stemmer est présenté et décrit dans le travail des chercheurs (Taghva, Elkhoury, & Coombs, 2005), intitulé Arabic stemming without a root dictionary (Lemmatisation arabe sans dictionnaire des racines) à l'institut de recherche en sciences de l'information de l'université du Nevada à Las Vegas (USA).

Le *ISRI Arabic Stemmer* partage de nombreuses caractéristiques avec le Khoja stemmer. D'où, la principale différence est que le ISRI stemmer n'utilise pas du dictionnaire des racines, dont des ajustements supplémentaires ont été apportés pour améliorer cet algorithme comme suivant:

1- Ajout de 60 mots vides.

2- L'algorithme commence par la suppression des signes diacritiques représentant les voyelles.

3- Ajout du schème (Patron) (تفاعيل) à l'ensemble des schèmes.

4- Normalisation tous les hamza de formes (إ, أ, آ) en Alif nu (ا), (après, cette étape est annulée car elle augmentait l'ambiguïté des mots et elle modifiait les racines d'origine.

5- Il supprime les préfixes de longueur de 2 ou 3 lettres, il supprime la conjonction 'et' (و) et puis il retourne une racine, si cette racine n'existe pas dans sa liste prédéfinie il renvoie une forme normalisée du mot original.

ISRI Arabic stemmer a été utilisé dans plusieurs applications liées à la RI basées sur du texte telle que la catégorisation des documents (Bsoul & Mohd, 2011). *ISRI Arabic Stemmer* est intégré dans la librairie gratuite NLTK[16] sous le langage Python .

III.3.2.7. MADAMIRA

MADA (**M**orphological **A**nalysis and **D**isambiguation for **A**rabic), *analyse morphologique et désambiguïsation pour l'arabe* est un utilitaire utilisé pour les textes bruts arabes, il examine tous les mots générés par l'analyseur morphologique arabe *Buckwalter* (*Buckwalter Arabic Morphological Analyzer –BAMA*[17]-) présenté par *LDC-Catalogue (Linguistic Data Consortium)* pour les applications de TAL, de RI et de traduction automatique. L'outil MADA utilise un analyseur morphologique, un algorithme SVM

[15] http://zeus.cs.pacificu.edu/shereen/research.htm
[16] https://www.nltk.org/_modules/nltk/stem/isri.html
[17] https://catalog.ldc.upenn.edu/LDC2004L02

(machine à vecteurs de support) et des modèles de langage N-gramme afin de produire pour chaque mot d'entrée, une liste de mots tokenisés couvrant différentes caractéristiques morphologiques, telles que le POS-Tag, le lemme, le genre, le nombre ou la personne.

AMIRA[18] est une boîte à outils développée pour l'analyse du texte arabe standard moderne (MSA), elle comprend un segmenteur de mots (*tokenizer*), un étiqueteur POS-Tag (*POS-Tagger*) et un segmenteur de phrase ou de syntagme (*Base Phrase Chunker -BPC-*).

Actuellement, MADA et AMIRA ont été fusionnés en une seule plateforme (Pasha et al., 2014) avec les caractéristiques distinctives les plus élevées pour le traitement de base du texte arabe: MADAMIRA[19], qui peut être aussi téléchargé via le Web[20]. MADAMIRA présente une interface entre les applications textuelles du TAL et l'algorithme d'apprentissage, d'où elle fournit également des sorties en format XML. MADAMIRA crée 11 modèles différents pour la tokenisation du texte, ces modèles sont caractérisés en termes de quels éléments sont tokenisés (segmentés) du mot original et dans quel format des tokens (mots) sont créés. MADAMIRA a présenté une performance de précision de tokenisation de 98,9% et de segmentation de 99,2% (Pasha et al., 2014). Dans la littérature la plateforme MADAMIRA a été utilisée dans plusieurs applications comme la RI multilingue, la reconnaissance des entités nommées et la traduction automatique.

III.3.2.8. Farasa

Farasa (signifie « aperçu » en arabe) est une boîte à outils (package complet) d'analyse rapide, précise et puissante pour le traitement du texte arabe. Farasa peut effectuer la segmentation, la lemmatisation, l'étiquetage POS-Tag, la diacritisation par seq2seq, l'analyse syntaxique par dépendances, l'analyse syntaxique par constituant, la reconnaissance des entités nommées et la vérification orthographique. Farasa[21] a été développé récemment par *Arabic Language Technologies Group* au *Qatar Computing Research Institute (QCRI)* à travers ses différents modules. En plus, cette API peut être utilisée par divers langages de programmation.

III.3.2.9. AraMorph

AraMorph est une implémentation Java de *BAMA (**B**uckwalter **A**rabic **M**orphological **A**nalyzer)*, le package est téléchargeable gratuitement en ligne [22]. AraMorph utilise des classes Java pour l'analyse morphologique des fichiers de texte codé en caractère arabe. Son principe, il se base sur la translittération des caractères arabes en caractères latins (Exemple le mot Lune -قمر - est translitteré à qmr), ensuite il utilise un algorithme aveugle pour décomposer le mot dans une séquence possible de préfixes, de racines et de suffixes,

[18] http://nlp.ldeo.columbia.edu/amira/index.php
[19] https://camel.abudhabi.nyu.edu/madamira/
[20] http://innovation.columbia.edu/technologies/cu14012_arabic-language-disambiguation-for-natural-language-processing-applications
[21] https://farasa.qcri.org/
[22] https://sourceforge.net/projects/aramorph/

puis il vérifie leurs présences dans des listes prédéfinies. L'outil AraMorph a été utilisé dans plusieurs applications de RI multilingue arabe/anglais, de Web mining et de l'arabe dialectal.

III.3.2.10. Stanford CoreNLP

Stanford CoreNLP[23] , c'est une plateforme implémentée sous Java présentée par (Manning et al., 2014), cette plateforme fournit plusieurs applications et outils de TAL tels que la radicalisation (stemming), l'étiquetage POS-Tag (POS-Tagging), la reconnaissance des entités nommées et l'analyse syntaxique. Elle a été développée initialement pour les textes bruts anglais, mais par la suite elle a été étendue pour prendre en charge d'autres langues à savoir ; l'arabe, le français, l'allemand et le chinois. La plateforme *Stanford CoreNLP* a été utilisée dans plusieurs applications liées à la recherche d'information comme l'extraction des syntagmes nominaux, la reconnaissance des entités nommées et l'étiquetage POS-Tag (POS-Tagging).

III.3.2.11. Stanford Parser

Stanford Parser[24] est un analyseur probabiliste en langage naturel implémenté en Java de l'université de Stanford, qui est utilisé pour analyser les structures grammaticales des phrases. Cet outil peut analyser les documents écrits en anglais, en arabe, en italien, en portugais, en bulgare et en chinois. Le modèle arabe de *Stanford Parser* a été entrainé (par apprentissage) sur les standards de *Penn Arabic Treebank (PATB)* version 3. L'outil prend en entrée le texte brut qui doit être d'abord segmenté avec le *Stanford Word Segmenter* pour l'arabe (mentionné dans la section III.3.2.1.) et puis il retourne la tokenisation, l'étiquetage POS-Tag (Pos-Tagging), la délimitation des phrases et l'analyse syntaxique.

III.3.2.12. AraNLP

La librairie AraNLP[25] est une boîte à outils téléchargeable gratuitement sur le web, elle est basée sur la plateforme Java pour le traitement du texte arabe présenté par (Althobaiti, Kruschwitz, & Poesio, 2014). Elle prend en charge les tâches les plus importantes de prétraitement qui peuvent aider les SRIs dans la phase d'analyse de la requête ou de l'indexation de la base documentaire, telle que la suppression des signes diacritiques et des ponctuations, la tokenisation, la segmentation des phrases, l'étiquetage POS-Tag (Pos-Tagging), la lemmatisation et la reconnaissance des entités nommées.

III.3.2.13. Penn Arabic Treebank (PATB)

Une *Treebank* (banque d'arbres) est une ressource linguistique composée de grands corpus comprennent des phrases annotées par des étiquètes syntaxiques et elles sont aussi vérifiées et validées manuellement par des experts. Ce type de corpus est très utile et serviable pour le développement de beaucoup d'applications de TAL, telles que la

[23] https://stanfordnlp.github.io/CoreNLP/
[24] https://nlp.stanford.edu/software/lex-parser.shtml
[25] https://sites.google.com/site/mahajalthobaiti/resources

tokenisation, la diacritisation, l'étiquetage POS-Tag, la désambiguïsation morphologique, la segmentation des syntagmes et des phrases, la reconnaissance des entités nommées et l'étiquetage sémantiques. Le *Penn Arabic Treebank (PATB)* a été développé par (Maamouri et al., 2004) pour soutenir les activités de recherche linguistique sur l'arabe standard moderne (MSA). La *PATB* a été utilisé comme une collection pour faire l'analyse syntaxique du texte arabe (Tounsi, Attia, & van Genabith, 2009) et pour entraîner l'étiqueteur POS-Tagger par l'apprentissage automatique. En RI, une *Treebank* peut être un outil très utile pour analyser et faire correspondre (matching) une requête de type exacte (option de la requête lorsque les mots-clés sont écrits entre guillemet) avec des phrases dans le document.

III.3.2.14. Fassieh

Fassieh est un outil d'annotation développé par RDI[26], il a été créé par (Mohamed Attia, Rashwan, & Al-Badrashiny, 2009) spécifique pour le texte arabe, il prend des grands corpus de textes arabes bruts et il produit des textes structurés avec certains types d'annotations tels que ; l'analyse morphologique arabe, l'étiquetage POS-Tag (POS-Tagging) arabe, l'étiquetage sémantique du lexique. Fassieh permet aussi la diacritisation, la transcription phonétique, la reconnaissance du texte arabe et sa séparation depuis d'autres langues et la conversion de certaines expressions verbales d'autres langues vers l'arabe avec une précision élevée. Fassieh a été utilisé aussi dans les applications de question/réponse et de RI pour l'analyse des requêtes et de génération des mots-clés utiles pour la reformulation des requêtes par des ressources sémantiques ou linguistiques externes.

III.3.2.15. Reconnaissance des entités nommées

Il existe peu d'outils disponibles gratuits pour la reconnaissance des entités nommées pour la langue arabe. Sauf certaines tentatives individuelles ont été faites pour développer ces outils, bien qu'ils ne soient pas disponibles pour un usage public, tels que ; ANERsys, l'outil de reconnaissance des entités nommées crée par (Benajiba, Rosso, & Benedíruiz, 2007) basé sur l'entropie maximale, dont il fournit un corpus (ANERcorp) contenant 150.000 mots (tokens) ou LingPipe[27], un autre outil utile pour les tâches de reconnaissance des entités nommées et d'annotation POS-Tag. Une autre ressource d'entités nommées pour l'arabe gratuite[28] est disponible en ligne, elle a été créée par (Mohammed Attia, Toral, Tounsi, Monachini, & van Genabith, 2010), totalisant 45.202 d'entités nommées, ces entités sont extraites de Wikipédia arabe et elles sont fournies avec une traduction en anglais et des informations ontologiques. Très récemment, l'institut de recherche en informatique de Qatar a publié plusieurs ressources gratuite et outils[29], supportant ainsi la langue arabe y compris la reconnaissance des entités nommées.

[26] https://rdi-eg.ai/ RDI (Development of Digital Systems), connu sous "The Engineering Company for the Development of Digital Systems"
[27] http://www.alias-i.com/lingpipe/
[28] https://sourceforge.net/projects/arabicnes/files/
[29] https://alt.qcri.org/resources/ et https://alt.qcri.org/tools/

III.3.3. Ressources sémantiques

Les ressources sémantiques peuvent être des ressources linguistiques comme les corpus, des bases sémantiques, des thésaurus, des ontologies ou des wikis. Nous présentons dans ce qui suit certaines de ces ressources sémantiques les plus utilisées dans le domaine de la recherche d'information.

III.3.3.1. WordNet

La ressource du WordNet a été détaillée dans le chapitre II précédent. En bref, c'est une ressource sémantique très importante pour la langue anglaise sous forme d'une base de connaissances lexicales, son architecture est fondée sur des concepts sous forme des synsets hiérarchisées principalement par la relation de généralisation hyperonyme (est-un 'is-a') et la relation holonyme (partie-de 'part-of'). Cette ressource est largement utilisée dans les travaux liés à la recherche d'information, tels que, la reformulation sémantique des requêtes et l'indexation conceptuelle de la base documentaire, ainsi la désambiguïsation sémantique des lexiques et la construction des bases lexicales des entités nommées.

III.3.3.2. Arabic WordNet

Arabic WordNet (AWN) ou le Wordnet arabe est une ressource sémantique concernant la langue arabe construite par (Black et al., 2006), sa création est basée sur la correspondance entre les sens des mots en arabe et ceux en anglais du WordNet de l'université de Princeton. En plus des synsets, L'AWN fournit aussi des informations concernant des entités nommées, les verbes et les noms. Elle permet également l'exploration des concepts en utilisant soit des mots (avec ou sans signes diacritiques) ou bien des racines, dont elle renvoie en retour tous les synsets contenants ces éléments recherchés.

AWN a gagné une popularité dans la communauté des chercheurs, d'où elle a été utilisée pour l'expansion des requêtes et la recherche sémantique. La synonymie, l'hyponymie, l'hypernymie sont les relations les plus utilisées. La dernière version d'AWN se compose de 23.481 mots (dont 13.808 mots arabes non discrétisés) en formant 11.269 synsets. Le WordNet arabe est téléchargeable gratuitement sur le web[30].

III.3.3.3. Arabic Wikipedia

Le Wikipédia arabe (*Arabic Wikipedia*) est la version en langue arabe de la bibliothèque ouverte de Wikipédia. Il a commencé en 2003, en avril 2021, il comptait plus de 1,2 Million d'articles et 2,1 Million d'utilisateurs enregistrés. Il s'agit de la 16eme plus grande édition de Wikipédia par nombre d'articles, et il se classe 8eme en termes de profondeur parmi les Wikipédias.

Les pages de Wikipédia ont été aussi utilisées par les chercheurs dans les systèmes de RI, de question/réponse pour la langue anglaise. Ces pages peuvent être aussi utilisées pour la classification des requêtes, des sujets (Topics), des questions et la reformulation des requêtes.

[30] http://globalwordnet.org/resources/arabic-wordnet/awn-browser/#BrowserDownload

III.3.3.4. DBpedia

DBpedia[31] est un projet communautaire créé pour extraire des informations dérivées de Wikipédia et les rendre disponibles sur le Web, avec des formats structurées, normalisées et adaptées au web sémantique. DBpedia vise aussi à interconnecter le contenu de Wikipédia avec d'autres bases de données ouvertes provenant du Web, dans le cadre de l'émergence du web vers l'open data. DBpedia permet ainsi aux utilisateurs d'interroger sémantiquement les relations et les propriétés des ressources Wikipédia, y compris des liens vers les autres bases de données connexes.

La version 2016-04 de DBpedia inclut 6 millions d'entités, dont 5,2 millions sont classées dans une ontologie cohérente, y compris 1,5 million de personnes, 810 000 de lieux, 135.000 d'albums de musique, 106.000 de films, 20.000 de jeux vidéo, 275.000 d'organisations, 301.000 d'espèces, 5.000 de maladies et des milliers d'institutions (dont 49.000 sociétés et 45.000 établissements d'enseignement).

DBpedia a réalisé une version en 119 langues, dont l'arabe. En raison de sa richesse en information structurée, DBpedia est une source très fiable pour créer des ressources sémantiques à utiliser ultérieurement dans l'indexation sémantique ou la reformulation des requêtes pour les SRIs.

III.3.4. Moteurs de recherche de test

Il existe plusieurs plateformes de moteurs de recherche ou des SRIs qui peuvent être utilisés pour la langue arabe dans les taches d'indexation et l'analyse des documents, l'analyse des requêtes, l'implémentation des nouvelles techniques et des nouveaux modèles de recherche. Dans cette section, nous présentons les outils les plus populaires disponibles pour ces importantes tâches.

III.3.4.1. Lucene

Lucene[32] est une bibliothèque de moteur de recherche des documents textuels développée par la fondation Apache[33], elle est multiplateforme et open source, entièrement écrite en Java. Lucene contient plusieurs classes pour effectuer des analyses de textes arabes, à savoir ; la classe *ArabicNormalizationFilter* qui est utilisée souvent pour la normalisation orthographique arabe et l'indexation des documents, et la classe *ArabicStemFilter* qui est employée pour la segmentation et la tokenisation des mots arabes. Dans la littérature plusieurs travaux ont utilisé Lucene pour faire des tests et d'expérimentation de leurs approches, pareillement nous l'avons aussi utilisé pour évaluer notre approche présentée dans le chapitre IV et dans notre expérimentation dans le chapitre V.

[31] https://www.dbpedia.org/
[32] https://lucene.apache.org/core/
[33] https://www.apache.org/

III.3.4.2. JIRS

JAVA Information Retrieval System (JIRS)[34] est un système de RI basé sur des passages, développé par (Gómez, Buscaldi, Rosso, & Sanchis Arnal, 2007). Ce SRI a l'avantage de pouvoir passer aux différents modules du système et leurs fonctionnalités modifiant la configuration du fichier XML. JIRS est un système indépendant de la langue qui peut être modifié pour la langue arabe afin de développer des SRIs ou des systèmes de question/réponse. JIRS comprend également un module de recherche de passage basé sur un modèle de densité à distance qui donne plus de poids aux passages des réponses dans le texte où les questions semblent plus proches.

III.3.4.3. Whoosh

Whoosh[35] est une bibliothèque de moteur de recherche, rapide, implémentée sous le langage Python, elle est multiplateforme, c'est-à-dire elle fonctionne où Python s'exécute sans nécessiter de compilation, sa dernière version est Whoosh 2.7.4[36] . Whoosh permet l'indexation et la recherche rapide dans une base documentaire, dans laquelle, les programmeurs peuvent l'utiliser pour ajouter facilement des fonctionnalités de recherche à leurs applications. Chaque élément du fonctionnement de Whoosh peut être étendu ou remplacé pour répondre exactement aux différents besoins. Par défaut, Whoosh utilise la fonction de classement *Okapi BM25F*, mais elle peut être facilement personnalisée. De plus, tout le texte indexé dans Whoosh doit être sous format Unicode et son principal avantage est qu'elle crée des index assez petits par rapport à de nombreuses autres bibliothèques de RI. L'indexation des textes Unicode permet donc d'indexer et de rechercher les documents de texte arabe. Whoosh est utilisé dans notre expérimentation dans le chapitre V.

III.3.4.4. Hibernate Search

Hibernate ORM est un outil de mappage objet-relationnel (Object–Relational Mapping) pour le langage de programmation Java. Il fournit un Framework (outil) pour mapper un modèle orienté objet vers une base de données relationnelle. Sa principale caractéristique est le mappage des classes Java vers les tables de base de données et le mappage des types de données Java vers les types de données SQL. Hibernate ORM offre également des fonctions de requêtes et de recherche.

Par ailleurs, quand ils existent des difficultés d'intégration d'un moteur de recherche en texte intégral dans une application Java, centrée sur un modèle comme le cas de Lucene. Ces difficultés peuvent être de :

- non-concordance structurelle : Comment convertir le domaine d'objet en index de texte seulement; comment gérer les relations entre les objets dans l'index.

[34] https://sourceforge.net/projects/jirs/
[35] https://pypi.org/project/Whoosh/
[36] https://whoosh.readthedocs.io/en/latest/index.html

- non-concordance de synchronisation : Comment garder la base de données et l'index synchronisés en permanence.
- inadéquation de la récupération : Comment obtenir une intégration transparente entre les méthodes de recherche de données centrées sur le modèle de domaine et la recherche en texte intégral.

Hibernate Search[37] vient comme un projet de moteur de recherche pour résoudre ces incohérences, il exploite les technologies Hibernate ORM et Apache Lucene, en offrant la possibilité d'effectuer des requêtes de recherche en texte intégral. Hibernate Search est un logiciel libre distribué sous la licence publique générale limitée GNU. Nous avons aussi utilisé Hibernate pour l'expérimentation dans le chapitre V.

III.3.5. Plateformes et environnements de développement linguistique

III.3.5.1. GATE

GATE[38] est un environnement de développement intégré (IDE), gratuit et open source pour effectuer des tâches de traitement du texte et le développement des outils de recherche d'information et de TAL. GATE a été aussi utilisé pour le développement des systèmes de question/réponse, d'extraction d'information, d'apprentissage d'ontologie, d'annotation de corpus et d'autres tâches de TAL. GATE fournit également des plugins pour traiter de nombreuses langues hors l'anglais telles que l'arabe, l'hindi, le français et l'allemand.

III.3.5.2. Nooj

Nooj[39] est un environnement de développement linguistique, développé par (Silberztein, Váradi, & Tadić, 2012), cet outil permet de créer et maintenir des ressources lexicales à large couverture, ainsi que des grammaires morphologiques et syntaxiques. Il comprend des dictionnaires et des grammaires qui peuvent être utilisés pour localiser des patrons (patterns) morphologiques, lexicaux et syntaxiques, et pour étiqueter des mots simples et composés. NooJ peut être facilement adopté pour la tâche de la reconnaissance des entités nommées afin d'identifier les noms des personnes et des lieux, les dates, les expressions techniques de la finance, etc. La plateforme Nooj se compose de trois modules ; la gestion du corpus en multi-formats, la création du lexique et le développement de la grammaire. Le module de gestion de corpus fournit des fonctionnalités de base telles que ; la collection des textes, l'indexation, l'étiquetage POS-Tag, l'annotation syntaxique, la désambiguïsation sémantique des mots et la recherche dans le corpus par des requêtes. Le module lexical permet à prédire des variantes orthographiques et des mots inconnus à l'aide de son module dictionnaire et de sa grammaire morphologique intégrée. Le module syntaxique permet à l'utilisateur de représenter les grammaires en langage naturel sous

[37] http://hibernate.org/
[38] https://gate.ac.uk/
[39] http://www.nooj-association.org/

forme de graphes de transition à états finis et de les appliquer aux corpus sous forme de requête. La plateforme multilingue Nooj prend en charge plus de 20 langues, dont l'arabe. Evidemment, Nooj a été utilisé par plusieurs éditeurs de logiciels informatiques pour construire des applications d'extraction d'information et des moteurs de recherche d'information.

III.3.6. Collections de test pour la recherché d'information arabe

Malheureusement, Ils existent très peu de collections pour tester et évaluer des SRIs ou des moteurs de recherche pour la langue arabe, et encore, la plupart ne sont ni gratuites ni disponibles en ligne par une licence accès libre (*open source)*. Dans cette section nous mentionnons ces rares collections.

III.3.6.1. Collection « LDC »

Nous citons dans ce qui suit les collections fournies par LDC[40] (Linguistic Data Consortium) de l'université de Pennsylvanie (USA) utilisées par quelques travaux dans la littérature. LDC est un créateur et distributeur d'un large éventail de ressources linguistiques et qui ne sont pas éventuellement en accès libre.

III.3.6.1.1. Pour la recherche ad hoc en arabe et multilingue

LDC Arabic Newswire Part 1 : la collection (LDC2001T55), qui contient 383.872 documents de presse de l'AFP (Agence France Presse):

1. Collection:

https://catalog.ldc.upenn.edu/LDC2001T55

2. Sujets (Topics) et jugements de pertinence:

a-TREC 2002 par sujets (topics) en multilingue en arabe:
http://trec.nist.gov/data/topics_noneng/CL.topics.arabic.trec11.txt

b- TREC 2002 par sujets (topics) en multilingue en anglais:
http://trec.nist.gov/data/topics_noneng/CL.topics.english.trec11.txt

c- TREC 2001 par sujets (topics) en multilingue en arabe:
http://trec.nist.gov/data/topics_noneng/arabic_topics.txt

d- TREC 2001 par sujets (topics) en multilingue en anglais:
http://trec.nist.gov/data/topics_noneng/english_topics.txt

e- TREC 2001 par sujets (topics) en multilingue en français:
http://trec.nist.gov/data/topics_noneng/french_topics.txt

[40] https://www.ldc.upenn.edu/

3. jugements de pertinence:

a- TREC 2002 qrels:
http://trec.nist.gov/data/qrels_noneng/qrels.trec11.xlingual.txt

b- TREC 2001 qrels:
http://trec.nist.gov/data/qrels_noneng/xlingual_t10qrels.txt

III.3.6.1.2. Détection des topics et le suivi (*Topic Detection and Tracking –TDT*)

1. Collection des documents:

a- TDT3: Un sous-ensemble de la collection *The LDC Arabic Newswire Part 1* (LDC2001T55) :
https://catalog.ldc.upenn.edu/LDC2001T55

b- TDT4 : Textes multilingues et les annotations:
https://catalog.ldc.upenn.edu/LDC2005T16

c- TDT5 : sujets (topics) et les annotations:
https://catalog.ldc.upenn.edu/LDC2006T19

2. Sujets (topics) et jugement de pertinence:

a- TDT5 inclut 10.000 sujets (topics) de jugement de pertinence :
https://catalog.ldc.upenn.edu/LDC2006T19

III.3.6.2. Collection « ZAD »

ZAD est une collection de test de RI créée par (Darwish & Oard, 2002), elle a été construite à partir de livre Zad Al-Mead[41], composé de 2.730 documents de type document image (à partir d'une copie imprimée du livre) et puis elle a été numérisée par *OCR* la reconnaissance optique des caractères *(OCR **O**ptical **C**haracter **R**ecognition)*. ZAD est développée sur 25 sujets (topics), le nombre de documents pertinents par sujet (topic) varie de zéro à 72 documents, avec une moyenne de 18 documents. La longueur moyenne des requêtes est de 5,5 mots. ZAD est disponible gratuitement, qui peut être utilisée pour évaluer des techniques alternatives de recherche de documents scannés du texte arabe.

[41] Zad Al-Mead est un livre écrit par le savant musulman Ibn al-Qayyim au sujet de Sira
زاد المعاد في هدي خير العباد كتاب من تأليف ابن قيم الجوزية في خمسة مجلدات، يتناول الفقه وأصوله والسيرة والتاريخ وذكر فيه سيرة الرسول صلى الله عليه وسلم.

III.3.6.3. Collection « KUNUZ »

KUNUZ est une collection[42] de test de RI polyvalente, composée de documents des textes arabes classiques structurés avec des voyelles. La collection KUNUZ a été créée par (Bounhas & Guirat, 2019), elle est composée de 7.031 documents extraits du *hadith*[43] *de Sahih al-Bukhari* structurés sous format XML (le livre original de Sahih al-Bukhari contient 7.563 hadiths, parce que les hadiths similaires sont regroupés dans KUNUZ). La collection contient ainsi 34 requêtes.

III.4. Recherche d'information pour la langue arabe

Les caractéristiques du texte arabe (section III.2) créent une ambiguïté accrue et complexe surtout pour les tâches telles que le prétraitement du texte, l'indexation automatique, l'analyse de la requête, l'extraction d'information et la reconnaissance des entités nommées.

Plusieurs travaux dans la littérature ont été introduits pour résoudre et améliorer les performances des SRIs pour l'arabe, ces travaux sont intervenus soit au niveau de la morphologie du texte et l'indexation soit au niveau de la requête, bien que d'autres études se sont intéressées à la RI multilingues en arabe (Z. Chen & Eickhoff, 2021) ou par identification des topics (El Kah & Zeroual, 2021), comme les études de (Harrag, Hamdi-Cherif, Al-Salman, & El-Qawasmeh, 2009, 2011) qui ont expérimenté les effets de certaines taches telles que la segmentation thématique qui peuvent avoir sur les performances de la RI arabe, où l'expérimentation a été réalisée sur une base de données de Traditions Prophétiques (Hadiths).

III.4.1. Morphologie du texte et indexation

III.4.1.1. Par morphologie

En raison de la complexité morphologique de la langue arabe, des premières études (Abu-Salem, Al-Omari, & Evens, 1999; Al-Kharashi & Evens, 1994; Bounhas & Guirat, 2019; Hmeidi, Kanaan, & Evens, 1997) se sont concentrées sur l'effet de la morphologie de la langue standard. Le but est de fusionner des mots proches et de significations similaires. Ces études ont suggéré que l'indexation du texte à l'aide des racines (roots) améliore l'efficacité de la recherche par rapport à l'utilisation de mots ou de radicaux (lemmes). Cependant, la morphologie dans de nombreuses de ces études a été effectuée manuellement. Une autre étude était effectuée sur un grand corpus par (Aljlayl et al., 2001) avait également montré que l'utilisation de la lemmatisation légère des mots, par la suppression des préfixes et des suffixes dans le texte du corpus pour l'indexation automatique pourrait améliorer les performances du SRI. D'autres travaux similaires (A. Chen & Gey, 2002; Darwish & Oard, 2002; Larkey, Ballesteros, & Connell, 2002) ont aussi suggéré que le prétraitement des textes

[42] http://jarir.tn/kunuzcorpus

[43] *Hadith* : recueil des actes et paroles du prophète Mohammed (sws) et de ses compagnons, à propos de commentaires du saint Coran ou de règles de conduite.

par la lemmatisation légère donne un meilleur résultat par rapport à l'analyse morphologique. (Darwish & Oard, 2002) a signalé que les défauts de l'analyse morphologique pourraient être attribués à des problèmes de couverture et d'exactitude. En ce qui concerne la couverture, les analyseurs échouent généralement à traiter les mots arabisés ou translittérés, qui peuvent avoir des préfixes et des suffixes attachés et qui doivent être indexés. Quant à l'exactitude, la présence (ou l'absence) d'un préfixe ou d'un suffixe peut modifier considérablement l'analyse du mot. Cependant une étude ultérieure de (Darwish & Ali, 2012) suggère que l'utilisation de la ressource d'AMIRA[44] pour la génération des lemmes (stems) similaires et l'analyse des pluriels brisés améliore encore l'efficacité des SRIs par rapport aux autres approches en raison du performance de la lemmatisation d'AMIRA et sa plus large couverture.

En revanche, dans (Guirat, Bounhas, & Slimani, 2016), les auteurs ont proposé une meilleure approche hybride combinant les deux unités, le lemme et la racine (stem et root) afin de créer l'index de manière à en tirer les avantages de chacun et à tenter de surmonter leurs lacunes, le principe de l'approche est d'attribuer un poids à chaque unité d'indexation et essayer de trouver les meilleures valeurs de pondération, plus tard les mêmes auteurs (Guirat, Bounhas, & Slimani, 2019) ont proposé l'optimisation de cet index hybride par les techniques de lissage pour attribuer des poids aux différents paramètres lors de la phase de pré-indexation, l'expérimentation menée sur la collection de test standard de LDC (*"Arabic Newswire Part 1", catalog LDC2001T55*) a montré une amélioration dans les performances.

III.4.1.2. Par N-grammes de caractères

L'utilisation des tri-grammes et les quadri-grammes dans la phase d'indexation a réalisé de bons résultats par rapport à la lemmatisation légère. La technique a été présenté dans les travaux menés par (Darwish & Oard, 2002; Mayfield, McNamee, Costello, Piatko, & Banerjee, 2002), cette dernière est utile puisque la longueur moyenne estimée d'un lemme (stem) arabe est d'environ 3 à 6 caractères. De plus, les caractères peuvent être aussi efficace parce que:

- Ils correspondent toujours à des lemmes de mots.
- Ils ne sont pas liés par un vocabulaire prédéfini comme l'analyse morphologique.
- Les n-grammes qui incluent des préfixes et des suffixes apparaissent plus souvent que les n-grammes qui incluent des lemmes, et par conséquent, le fichier index rétrograderait automatiquement les poids des n-grammes qui ont des préfixes et des suffixes et il favoriserait les poids de n-grammes qui incluent des lemmes. C'est comme les préfixes et les suffixes deviennent similaire à des mots vides.

En revanche, le principal inconvénient de cette technique est qu'elle augmente considérablement la taille de l'index et elle nécessite plus de calcul pour le traitement, des solutions ont été apportées par la suppression des signes de diacritisation et la normalisation des caractères.

[44] http://nlp.ldeo.columbia.edu/amira/index.php

III.4.1.3. Par ressource

Plus récemment, les travaux de (El Mahdaouy, El Alaoui, & Gaussier, 2018) ont démontré que les modèles de langage basés sur le plongement des mots (Word-Embedding) surpassent considérablement l'approche d'indexation sémantique basée sur WordNet arabe. Cela s'explique par le fait que la ressource WordNet arabe (AWN) est limitée pour couvrir toute la langue arabe et lorsqu'elle a été utilisée seule comme une ressource externe.

III.4.2. Reformulation de requêtes

Plusieurs techniques ont été proposées, nous citons, les travaux dans (Shaalan, Al-Sheikh, & Oroumchian, 2012), dans lesquels les auteurs ont proposé une méthode d'expansion des requêtes basée sur la similarité des termes pour améliorer la RI arabe en utilisant une technique appelée Attente-Maximisation qui ajoute des termes similaires des mots-clés à la requête originale. Leurs expériences ont montré que la technique améliore généralement la précision en récupérant plus de documents pertinents. Les études de (Mallat, Zouaghi, Hkiri, & Zrigui, 2013) ont présenté un enrichissement des requêtes en se basant sur le contexte linguistique afin d'améliorer les performances des SRIs. Le but de la méthode est de générer une liste descriptive contenant un ensemble de lexiques linguistiques à attribuer à chaque terme significatif de la requête, cette liste est créée par des informations contextuelles issues du corpus. (Mahgoub, Rashwan, Raafat, Zahran, & Fayek, 2014) ont introduit une approche de reformulation des requêtes par des nouveaux termes en utilisant une ontologie construite à partir de pages Wikipédia afin d'améliorer la précision de la recherche en langue arabe. D'autres scénarios de reformulation de requêtes et d'indexation de documents pour la RI arabe ont été évalués sur la base d'un graphe de connaissances morpho-sémantique dans les travaux de (Bounhas, Soudani, & Slimani, 2020).

Récemment, le WordNet arabe (Arabic WordNet AWN) est également proposé comme une ressource externe pour améliorer la recherche d'information. De départ, c'était une base lexicale (avec des relations sémantiques) qui a été développée pour surmonter les problèmes d'analyse liés à la langue, elle est basée sur Princeton WordNet (PWN), qui comprend au total 23.481 mots et 11.269 synsets (Elkateb et al., 2006). L'AWN est souvent utilisée par les chercheurs comme une ressource pour la recherche d'information sémantique ou la recherche d'information basée sur les concepts, elle est employée pour la récupération de documents en fonction de la signification exprimée par les termes de la requête. Dans leurs approches, les chercheurs l'utilisent en l'associant avec différentes techniques d'expansion des requêtes. Nous citons les travaux en relation suivants. (Abderrahim, 2014) a examiné et comparé deux techniques d'expansion des requêtes pour la RI arabe, plus exactement son approche est basée sur les concepts utilisant le WordNet arabe (AWN) et la pseudo-réinjection de la pertinence (Pseudo Relevance Feedback -PRF-), cette technique a été déjà expérimentée pour l'anglais dans les compagnes d'évaluation de TREC, dans laquelle les résultats sont globalement améliorés. Les résultats de (Abderrahim, 2014) ont montré que la technique de PRF est meilleure que la recherche basée sur les concepts de la ressource d'AWN, ainsi la reformulation par la PRF peut améliorer d'environ 4% les performances du SRI pour l'arabe. (Abbache, Meziane, Belalem, & Belkredim, 2018) ont

décrit une autre méthode d'expansion automatique des requêtes dans le contexte de la langue arabe en utilisant le WordNet arabe (AWN) et les règles d'association, cette méthode est basée sur l'hypothèse que les mots qui ont tendance à apparaître ensemble dans les documents sont susceptibles d'avoir des significations similaires et ils peuvent donc être associés, ce qui a permis d'inclure ces nouveaux mots dans le processus final de reformulation de la requête.

Contrairement aux travaux existants, notre approche présentée dans le chapitre IV de cette thèse propose de combiner au même temps, des connaissances du WordNet arabe et la technique de pseudo-réinjection de la pertinence (PRF) afin de créer un arbre sémantique représentant la requête.

III.5. Conclusion

Au début de ce chapitre nous avons présenté en bref les caractéristiques du caractère, du mot et du texte de la langue arabe. Dans la suite et au long de ce chapitre, nous avons vu l'état de l'art sur les différents outils et ressources pour le développement des applications et des solutions concernant l'analyse de la langue, pour lesquels, nous avons exposé les bibliothèques, les packages et les plateformes les plus populaires dans les différents processus de l'analyse du texte (comme Khoja Stemmer, ISRI Stemmer, MADAMIRA, Farasa, AraNLP,…), les ressources sémantiques (WordNet arabe, Wikipedia arabe, DBpedia,), les plateformes et les environnements de développement du TAL (Gate, Nooj) et les moteurs de recherche acceptant le texte et la RI arabe (comme Lucene de Java ou Whooh de Python). Nous avons aussi cité les collections de test existantes pour la RI arabe à savoir ; LDC, ZAD et Kunuz. Enfin, nous avons présenté les travaux connexes ainsi leurs résultats qui ont déjà étudié la problématique de la recherche d'information pour la langue arabe. Dans les deux chapitres suivants nous présenterons nos contributions concernant, premièrement, l'amélioration de la RI arabe par la reformulation des requêtes en se basant sur l'arbre sémantique et deuxièmement, la création d'une collection de test de RI pour l'arabe.

Chapitre IV. Amélioration de la recherche d'information basée sur l'expansion sémantique des requêtes : application à la langue arabe

IV.1. Introduction

Au cours de ce présent quatrième chapitre, nous présentons notre première contribution, il s'agit d'une nouvelle approche pour concevoir une technique de reformulation des requêtes basées sur l'arbre sémantique. Cette nouvelle technique conforte à améliorer les performances des systèmes de recherche d'information pour la langue arabe.

Le chapitre est organisé comme suit ; tout d'abord, la section IV.2 expose un aperçu sur la partie théorique de la technique proposée, cette dernière est subdivisée en trois principales étapes : étape 1) prétraitement de la requête, étape 2) l'extraction et la pondération des concepts relatifs aux mots-clés de la requête et finalement l'étape 3) qui sert à la construction de l'arbre sémantique interprétant le besoin de l'utilisateur.

Afin de tester l'efficacité de la technique proposée, la section IV.3 présente la réalisation de l'expérimentation sur une collection de test de recherche d'information avec un vrai ensemble de requêtes, dont nous examinons en détails les résultats expérimentaux obtenus, ensuite, nous présentons une discussion sur ces résultats dans la section IV.4, dans laquelle nous mentionnons les limites et les obstacles ainsi que les points forts de cette nouvelle technique. Enfin, le chapitre se termine par une conclusion sur l'aboutissement de l'approche proposée.

IV.2. Approche proposée

Nous savons bien qu'un système de recherche d'information (SRI) est composé d'une partie pour l'indexation de la base documentaire, une partie pour la représentation et l'amélioration de la requête et une partie pour le modèle d'appariement entre l'index et la requête (Chapitre I). L'objectif de notre approche est de proposer une nouvelle technique pour la représentation et la reformulation de la requête afin d'améliorer le processus général de la recherche appliqué pour l'arabe. De ce fait, nous proposons une technique de reformulation (expansion) de requête combinant deux méthodes, une méthode basée sur l'approche locale et une méthode basée sur une ressource externe. Plus précisément, nous combinons la méthode de pseudo-réinjection de la pertinence (en anglais, PRF Pseudo Relevance feedback) et la ressource WordNet arabe pour représenter le besoin de l'utilisateur par un arbre sémantique. La Figure IV.1 suivante schématise l'idée générale proposée.

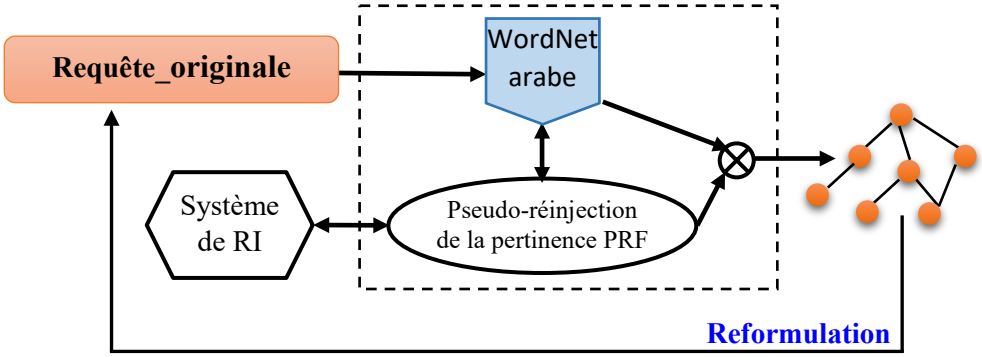

Figure IV.1. Principe de reformulation de la requête par l'arbre sémantique.

Le processus est donc complètement indépendant du SRI, c'est-à-dire que le processus d'expansion ne modifie pas ni l'index des documents ni le modèle d'appariement du SRI, mais de reformuler la requête originale à une requête plus riche et plus précise afin de se rapprocher le plus possible au besoin de l'utilisateur. Plus en détail, le processus commence par une étape de prétraitement de la requête originale pour extraire des concepts. Dans la deuxième étape, les termes de ces concepts vont être envoyés au SRI afin de récupérer les k-top documents par le principe de PRF, les termes de la liste de ces documents récupérés sont aussi utilisés pour trouver des nouveaux concepts. Dans la troisième étape, nous construisons l'arbre sémantique à l'aide du WordNet arabe (AWN) pour représenter la requête, cet arbre est alors composé par des concepts trouvés dans les deux étapes précédentes. La Figure IV.2 illustre le processus de l'approche à travers ses principales étapes.

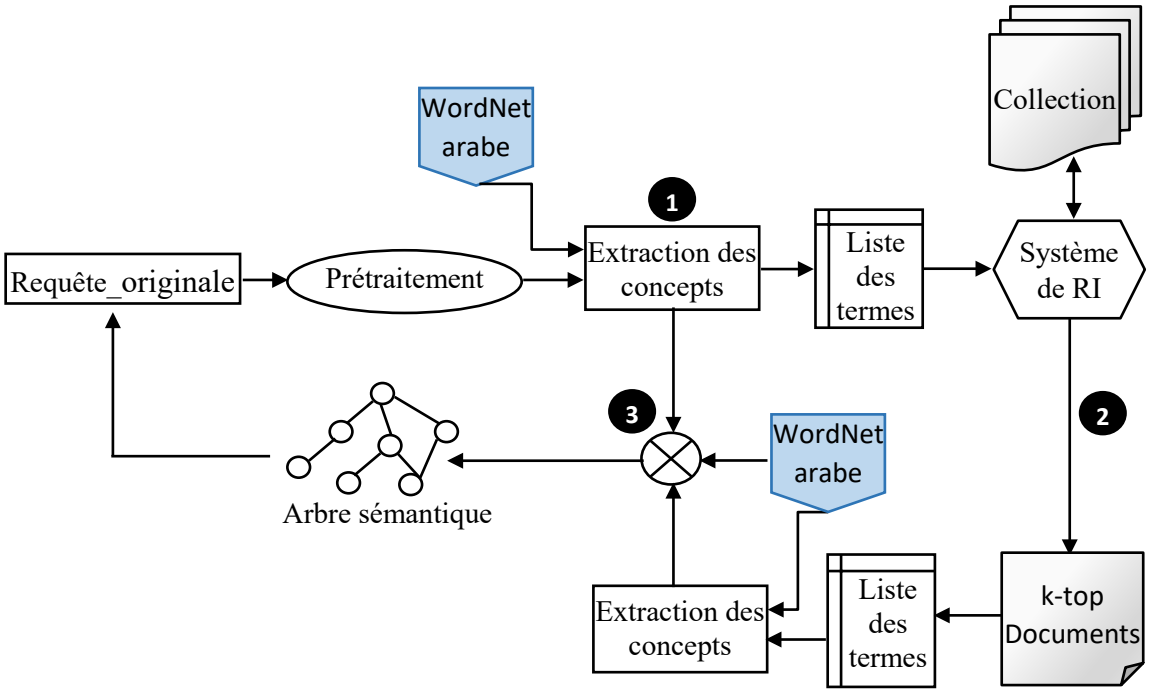

Figure IV.2. Etapes de l'approche proposée.

IV.2.1. Etape 1 : Prétraitement et extraction des concepts

L'étape de prétraitement est une phase indispensable utilisée par les SRIs afin de récupérer les termes et les mots qui vont être utilisés dans la phase de la recherche dans l'index. Dans notre approche, nous avons pris en considération les caractéristiques spécifiques linguistiques et morphologiques de l'arabe (vu dans le Chapitre III), alors nous avons réadapté et réutilisé les techniques habituelles servies dans les applications de TAL comme suit :

IV.2.1.1. Segmentation

Il s'agit de trouver les mots et les termes de base à partir de la requête originale écrite en langage naturelle, de ce fait, nous utilisons les délimiteurs représentés par la ponctuation, les caractères non alphabétiques et les blancs comme séparateurs. Le résultat de cette phase est une liste des mots simples.

IV.2.1.2. Normalisation

La normalisation sert à regrouper les mêmes mots qui s'écrivent différemment à cause de la diacritisation, l'allongement des caractères arabes, les caractères spéciaux, les lettres étrangères (lettres d'autres langues comme le Latin) et les chiffres qui sont collés aux mots. En conséquence nous devons supprimer tous ces signes diacritiques, les chiffres et les caractères spéciaux, également nous devons remplacer (أ , إ et آ) par (ا) et la terminaisons pour les mots de longueur de plus de trois lettres : de (ي) par (ى) et (ة) par (ه). (Farghaly & Shaalan, 2009).

Exemple de normalisation

Mot originale arabe	en français	après remplacement
رأس	Tête	راس
إسم	Nom	اسم
آبل	Apple	ابل
موسيقة	Musique	موسيقه
دوري	Période	دورى

IV.2.1.3. Suppression des mots vides (Stopwords)

La technique de suppression des mots vides dans la requête est généralement utilisée par les SRIs pour ne garder que les mots-clés nécessaires utilisés dans la phase de recherche. Concernant la recherche d'information arabe, cette technique a été utilisée avec succès dans les travaux de (Chen & Gey, 2002; Xu, Fraser, & Weischedel, 2002). Pour la langue arabe, il existe une grande liste de ces mots vides en raison de la richesse du lexique, elle inclue évidemment certains liens grammaticaux tels que l'article défini "le" (ال), les prépositions attachées et séparées, les conjonctions, les mots interrogatifs, les mots négatifs, les exclamations, les lettres d'appel, les adverbes de temps et de lieu. Elle comprend également tous les pronoms, les démonstratifs, les pronoms sujet/objet, les cinq noms distinctifs, certains nombres, les additions et les verbes. Dans notre approche, nous utilisons la liste générale créée et testée par (El-khair, 2006). Elle a été collecté sur la base des caractéristiques de la langue arabe, cette liste générale contient un grand nombre de mots vides, elle est égale à 1.377 mots. Une problématique de certains mots vides qui sont attachés aux mots-clés soit de préfixes ou de suffixes, exemple le mot vide de conjonction de coordination (و / avec) est peut-être attaché au mot (رأس / tête) ce qui devient (ورأس /avec tête), donc l'identification des mots vides peut nécessiter d'abord la lemmatisation (Stemming) par la suppression des préfixes ou/et des suffixes.

IV.2.1.4. Lemmatisation

Les phénomènes morphologiques se subdivisent en deux groupes :

- **la flexion** : phénomènes purement grammaticaux (genre, nombre, personne, mode, temps) n'affectant pas la catégorie syntaxique.
- **la dérivation** : permet de créer de nouvelles unités lexicales.

Exemple :

Pour le verbe (entendre/ سمع), nous avons :

- **Formes flexionnelles** : (ils entendent / يسمعون) (nous entendons / نسمع) (j'entends / استمع) (j'ai entendu / سمعت).
- **Formes dérivationnelles** : (casque / سَمَّاعة) (audible / مسموع).

La lemmatisation (racinement ou stemming en anglais) est la technique de trouver le mot origine (lemme ou racine) pour les différentes formes morphologiques flexionnelles ou dérivationnelles, ayant des variations orthographiques. Dont, les lemmes sont souvent utilisés pour améliorer les performances des SRIs lors de la phase de recherche, car ils réduisent le nombre des variantes à un seul mot racine commun. Plusieurs outils et techniques sont proposés dans la littérature (Chapitre III), dans notre approche, nous avons optés d'utiliser le Khoja Stemmer [1] pour ses performances positives et sa simplicité, aussi c'est l'un des outils les plus utilisés pour la langue arabe (Khoja & Garside, 1999).

[1] http://zeus.cs.pacificu.edu/shereen/research.htm

IV.2.1.5. Extraction des termes

La langue arabe possède deux types de termes : les termes simples constitués d'un seul mot et les termes composés constitués de deux mots ou plus.

- **Termes simples**

Un terme simple est un seul mot isolé et non vide qui peut être utilisé comme mot-clé de la requête pour décrire le besoin de l'utilisateur. Initialement, tous les mots issus de la phase de prétraitement sont considérés comme des termes simples, et puis nous cherchons parmi eux les termes composés selon le principe suivant ;

- **Termes composés**

L'arabe possède aussi un grand nombre de termes composés par deux mots ou plus, ces termes composés font référence à des concepts indépendants, alors si ces mots sont pris séparés, ils ne signifient pas la même sémantique quand ils sont pris en ensemble.

Par exemple, le terme (dinde / الديك الرومي) qui est composé de deux mots simples q1= الديك et q2=الرومي . D'après le dictionnaire en ligne almaany[2], Il y a deux (02) sens pour q1 et dix (10) sens pour q2, donc un total de 2x10=20 combinaisons de sens possibles.

Dans cette phase, nous récupérons alors les termes composés en utilisant la technique de n-gramme des mots. Tout d'abord, la requête est entièrement considérée comme un seul terme composé, dont le paramètre N est égal au nombre de mots dans la requête, ensuite, à chaque fois le N est décrémenté par une valeur 1 (taille (requête) - 1), puis nous comparons ces nouveaux termes avec les termes existants dans la ressource WordNet arabe, quand un terme est trouvé dans la ressource, il est donc retenu comme un terme composé.

IV.2.1.6. Extraction et désambiguïsation des concepts

i) Identification des concepts

Dans cette étape, la liste des termes simples et des termes composés trouvée dans la phase précédente est vérifiée par rapport à la ressource du WordNet arabe, dans laquelle, nous examinons si les termes obtenus sont non ambigus, c'est-à-dire un terme doit appartenir à une seule entrée dans le WordNet arabe, ou bien plus précisément le terme se trouve dans un seul synset, dans ce cas, le synset associé est sélectionné et puis il est ajouté à la liste des concepts originaux. Autrement, si le terme obtenu est ambigu, cela signifie qu'il appartient à plus d'un synset alors ce terme doit être désambiguïsé, c'est-à-dire de trouver le bon concept (synstet) associé.

[2] https://www.almaany.com/

ii) Procédure de désambiguïsation

Pour la désambiguïsation des concepts (représentés par des synsets), nous devons calculer la similarité sémantique entre ces concepts et la requête originale. Plusieurs méthodes ont été proposées et utilisées par des auteurs dans leurs travaux de recherche, à savoir, la distance sémantique entre les concepts de l'ontologie et leurs scores de poids tels que les travaux présentés par (Baziz, Boughanem, & Traboulsi, 2005; Boubekeur, Boughanem, & Tamine-Lechani, 2007). Dans notre approche, nous proposons d'utiliser la formule (IV.1) du coefficient de Jaccard (Jaccard similarity coefficient), motivée par l'étude de (Pal, Mitra, & Datta, 2014), et puis nous calculons un score en utilisant la formule (IV.2). Le score calculé fait référence à la similarité entre chaque concept ambigu (tout en entier) et l'ensemble des termes de la requête originale.

$$Sim_{Jaccard}(t, q_i) = \left(\frac{n_{t,q_i}}{n_t + n_{q_i} - n_{t,q_i}} \right) \quad \text{(IV.1)}$$

La formule de Jaccard calcule la similarité entre deux termes. Où : *t* est un terme quelconque, q_i est un terme de la requête, n_t et n_{t,q_i} désignent, respectivement, le nombre de documents dans lesquels le terme *t* apparaît et le nombre de documents dans lesquels les termes *t* et q_i co-apparaissent.

$$Score(Cp) = \frac{1}{long(cp)} \left(N_{tc} + \sum_{i=1}^{long(cp)} Sim_{Jaccard}(t, s_i) \right) \quad \text{(IV.2)}$$

Où: *long(cp)* désigne le nombre de termes du synset *Cp*. N_{tc} désigne le nombre de termes communs entre le synset *Cp* et la requête originale, *t* désigne le terme à désambiguïser, s_i est un terme du synset *Cp*.

Ainsi, la procédure de la désambiguïsation passe par l'affectation du concept approprié à celui qui maximise ce score au terme ambigu recherché.

iii) Pondération des concepts

Nous savons bien que dans une requête il existe des termes plus importants par rapport à d'autres, étant donné que ces termes importants constituent l'idée centrale de la requête et le reste des termes constitue les éléments périphériques ou complémentaires.

La pondération des concepts sert à designer alors le/les concept(s) centrale(aux) par rapport aux concepts périphériques dans la requête. Par conséquent, différents poids doivent être attribués aux différents concepts en fonction de leurs importances dans la requête. Plusieurs méthodes ont été utilisées dans les travaux connexes, dans notre approche, nous utilisons le même score de poids calculé précédemment par la formule (IV.2) et nous l'affectons au concept recherché

IV.2.2. Etape 2 : Extraction des concepts par pseudo-réinjection de la pertinence

Dans cette étape, nous récupérons les nouveaux concepts retrouvés dans le résultat de recherche de PRF (pseudo-réinjection de la pertinence), il s'agit dans les K-top documents retournés par le SRI après l'opération de la première recherche effectuée sur les termes des concepts sélectionnés dans l'étape précédente.

IV.2.2.1. Extraction et pondération des termes

La technique de PRF (Pseudo Relevance Feedback) pseudo-réinjection de la pertinence a été largement utilisée dans les travaux de recherche en RI. Elle a été implémenté pour la reformulation des requêtes, dans les différents modèles de RI (modèle vectoriel, modèle probabiliste, etc.). Pareillement, le principe de PRF a été implémenté aussi dans la modélisation du langage (Cao, Nie, Gao, & Robertson, 2008).

Le principe de la technique de PRF est d'utiliser les termes des K-top documents retournés qui sont supposés pertinents dans le processus de la reformulation de la requête. Dans notre approche proposée, nous suggérons de sélectionner que les termes les plus représentatifs, avec laquelle, nous estimons cette représentativité par un poids calculé par la formule (IV.3) qui prend en compte leurs proportions de fréquences. Les termes qui ont des valeurs de poids dépassant un seuil (m) sont sélectionnés comme termes candidats. La valeur de m est désignée empiriquement.

$$P_i = \frac{freq_i}{Max\_freq} \qquad \text{(IV.3)}$$

Où : P_i est le poids du terme i, $freq_i$ désigne la fréquence du terme i dans les K-top documents retournés, Max_freq désigne la fréquence la plus élevée des termes dans les K-top documents retournés.

IV.2.2.2. Extraction et désambiguïsation des concepts

Après la sélection des termes les plus représentatifs, nous recherchons leurs concepts correspondants dans la ressource du WordNet arabe, pour cela nous appliquons la même procédure d'identification et de désambiguïsation présentée dans la section (IV.2.1.6. Extraction et désambiguïsation des concepts). Identiquement, la pondération de ces concepts est aussi calculée par la formule (IV.2).

IV.2.3. Etape 3 : Construction de l'arbre sémantique

Les résultats obtenus à la suite de l'étape 1 et l'étape 2 forment une liste de concepts. Cette liste de concepts est utilisée maintenant dans cette troisième étape pour construire l'arbre sémantique correspondant à la requête dans le but de capturer la sémantique du besoin de l'utilisateur. Les termes de cet arbre sémantique sont alors employés plus tard dans la phase de la reformulation de la requête

IV.2.3.1. Amorçage de l'arbre sémantique

Nous commençons d'abord par l'élimination des concepts doubles issus à la suite de l'étape 1 et l'étape 2 précédentes, puis un ensemble de pseudo-arbres sémantiques est généré à l'aide de la ressource du WordNet arabe (AWN), dont les concepts initiaux (C_i : les concepts liés aux termes de la requête originale plus (+) les concepts identifiés par PRF) et leurs synonymes (Cs_i) s'initialisent dans des racines. Ensuite, des nouveaux concepts sont ajoutés et étendus aux concepts initiaux selon la relation d'hyponymie à deux niveaux pour les concepts initiaux (C_i) et à un seul niveau pour les concepts synonymes (Cs_i) (nous proposons qu'un seul niveau pour les concepts synonymes pour ne pas s'éloigner du sens exprimé par le besoin de l'utilisateur à travers la requête originale). L'algorithme 01 suivant décrit le processus d'amorçage de l'arbre sémantique.

Algorithme 01 :

```
Début
Lire (Liste_des_concepts)
Tant que Non Fin_Liste_des_concepts Faire
    Début
        C← lire (un concept (Liste_des_concepts) )
        i←1
        j←1
        Nœud(i) ← C
        Cp ← C
            {Récupération des concepts hyponymes}
        Tant que Non Fin_hyponymes(AWN(Cp)) Faire
            {Récupération des concepts hyponymes Niveau I}
          Début
            C_h ← hyponyme (AWN(Cp))
            Nœud(ij) ← C_h
            Nœud(C_h) relation_fils Nœud(Cp)
            Cp_h ← C_h
            j++
            {récupération des concepts hyponymes Niveau II }
```

```
17:             Tant que Non Fin_hyponymes (AWN(Cp_h)) Faire
18:                 Début
19:                     C_h ← hyponyme (AWN(Cp_h))
20:                     Nœud(ij) ← C_h
21:                     Nœud(C_h) relation_fils Nœud(Cp_h)
22:                     j++
23:                 Fin_Tq
24:             j++
25:           Fin_Tq
26:       i++
27:       j←1
                {récupération des concepts synonymes}
28:       Tant que Non Fin_synonyme(AWN(Cp)) Faire
29:         Début
30:           C_s ← synonyme (AWN(Cp))
31:           Nœud(i) ← C_s
32:           Tant que Non Fin_hyponymes(AWN(C_s)) Faire
                {récupération des concepts hyponymes Niveau I pour les concepts synonymes }
33:             Début
34:               C_h ← hyponyme (AWN(C_s))
35:               Nœud(ij) ← C_h
36:               Nœud(C_h) relation_fils Nœud(C_s)
37:               j++
38:             Fin_Tq
39:         Fin_Tq
40:     Fin_Tq
41: Fin.
```

La Figure IV.3. suivante illustre le résultat de l'algorithme d'amorçage de l'arbre sémantique.

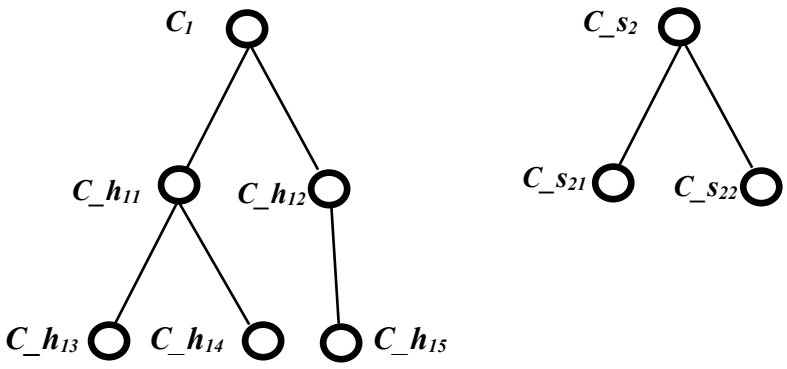

Figure IV.3. Amorçage de l'arbre sémantique.

IV.2.3.2. Extension et construction de l'arbre sémantique

Apres l'amorçage de l'arbre sémantique par la génération des pseudos branches de l'arbre à partir des concepts initiaux et leurs concepts synonymes, des nouveaux concepts sont extraits à partir de la ressource WordNet arabe (AWN) et ils sont ajoutés à ces branches de l'arbre, en partant vers le haut par l'utilisation de la relation sémantique ascendante d'hyperonymie [fils-père, soit par la relation ***Is-a*** (est-un) ou par la relation ***Part-of*** (partie-de)] jusqu'à arriver au nœud (concept) général commun, ce nœud (concept) représente alors la racine (concept racine *C_r*) de notre arbre sémantique intégré, comme le montre la Figure IV.4.

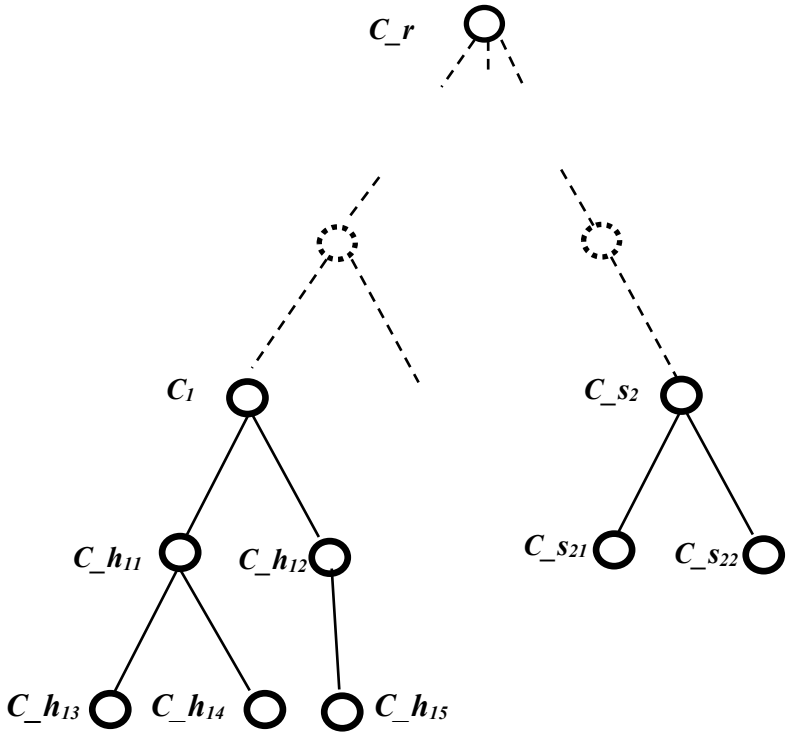

Figure IV.4. Construction de l'arbre sémantique**.**

Pour réaliser l'expérimentation pratique, nous avons utilisé la liste des requêtes[3] créée et mise en ligne par nous-même via l'actuel projet de thèse (le détail est présenté plus tard dans la section *IV.3.1. Collection de test*).

La Figure IV.5 en dessous illustre un exemple réel de génération de l'arbre sémantique pour la requête Q116 = [*le mouvement rebelle au Soudan* " حركة التمرد في السودان "].

```
<num>116</num>
  <title>حركة التمرد في السودان</title>
  <title_e>the rebel movement in Sudan</title_e>
```

[3] https://sourceforge.net/projects/queries-for-arabic-osac-corpus/

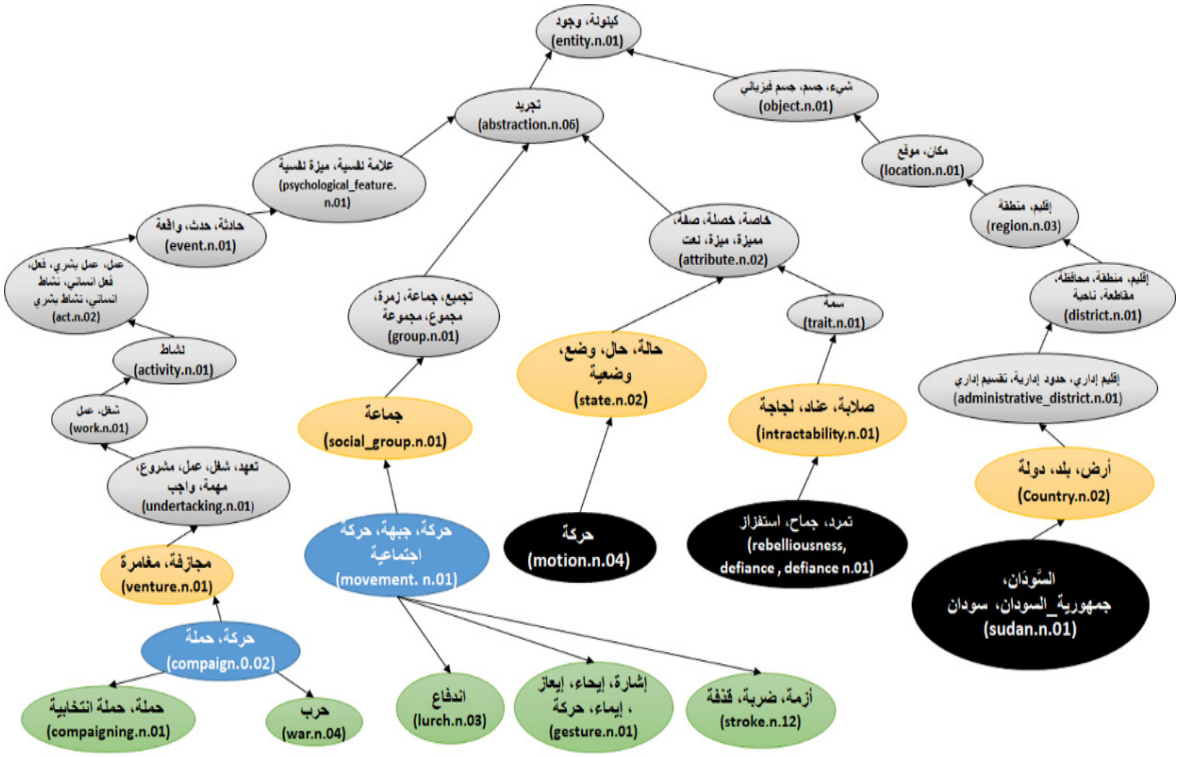

Figure IV.5. Arbre sémantique de la requête Q116.

IV.2.3.3. Pondération des nouveaux concepts et taillage de l'arbre sémantique

Afin de se maintenir sémantiquement près de la requête, nous devons pondérer les poids des nouveaux concepts et de ne garder que les nœuds (concepts) proches aux termes originaux de la requête.

Premièrement, nous calculons les poids des nouveaux concepts sur le principe de trouver une moyenne de similarité entre tout nouveau concept *(C_n)* et les concepts originaux (*C_o)* comme indique la formule (IV.4).

$$Poids(C\_n) = \frac{1}{k}\sum_{i=1}^{k} sim(C\_n \ , \ C\_o) \qquad \text{(IV.4)}$$

Où :

C_n désigne un nœud (concept) dans l'arbre sémantique à l'exception des concepts initiaux (Concepts originaux + concepts de PRF).

sim() est une fonction qui calcule les similarités sémantiques entres les concepts liés à la collection (puisque nous cherchons la similarité liée au contexte général dont la collection de RI a été créée et puis utilisée, or l'arbre sémantique reste uniquement liée à cette collection). Nous proposons la formule (IV.5) pour calculer cette similarité, dans laquelle, elle se base sur le principe de calculer le nombre de termes communs entre deux concepts

(les termes sont récupérés des synsets des concepts) divisé par la différence de fréquences de ces deux concepts dans la collection, plus en détail, si deux concepts ont presque les mêmes fréquences (ou ils sont égaux) dans une même collections, le nombre de termes en commun peut indiquer leur similarité sémantique, sinon, s'il existe une grande valeur de leurs fréquences dans la collection, cela indique que ces deux termes sont sémantiquement disjoints. Aussi, nous ajoutons une constante de valeur 0.5 pour le lissage da la formule et pour ne pas avoir la division par zéro.

$$sim(C_1, C_2) = \frac{Terme\_commn(C_1,C_2)+0.5}{|freq(C_1)-freq(C_2)|+0.5} \quad \text{(IV.5)}$$

Où :

Terme_commun (C_1, C_2) est le nombre de termes communs trouvés dans les sysnsets indiqués par les concepts C_1 et C_2 . *freq(C)* est calculée par la formule (IV.6).

$$freq(C) = \frac{1}{long(C)} \sum_{t=1}^{long(C)} fq(t) \quad \text{(IV.6)}$$

long(C) indique le nombre de termes dans le synset indiqué par *C*, *fq(t)* est la fréquence du terme *t* dans tous les documents de la collection.

Deuxièmement, après le calcul des poids des nouveaux concepts, nous taillons notre arbre sémantique par l'élimination des concepts sémantiquement distants aux concepts initiaux. Nous définissons l'algorithme 02 suivant pour décrire le processus de pondération des nouveaux concepts et le taillage de l'arbre sémantique.

- Soit *g* est le nombre des nœuds (concepts) dans l'arbre à l'exception des concepts initiaux.
- *S* représente le seuil minimum (déterminé expérimentalement = 0,5) à partir duquel, nous disons que deux concepts sont sémantiquement liés.

Algorithme 02 :

```
1:  Début
2:  Lire (Arbre)
3:  Pour i ← 1 à g Faire  {g = nombre de nœuds  dans l'arbre à l'exception des concepts initiaux}
4:      Début
5:        Lire (concept_arbre)        {sauf concept_initial}
6:        Tant que Non_Fin_concepts_initiaux Faire
7:          Début
8:            Lire (concept_initial)
9:            Si (Similarité(concept_initial ,concept_arbre)≥S) Alors
10:               assigne poids(concept_arbre)  comme poids dans Arbre
11:           Sinon
12:               supprime concept_arbre de Arbre
13:           Fin_Si
14:         Fin_Tq
15:     Fin_Pour
16: Fin.
```

IV.2.4. Processus de reformulation de la requête

Nous arrivons finalement à l'étape du processus de reformulation de la requête originale, il s'agit de l'expansion ou de la composition d'une nouvelle requête qui mieux représente sémantiquement le besoin de l'utilisateur, par l'ajout et la pondération des nouveaux termes récupérés depuis les concepts (Synsets) qui forment l'arbre sémantique. La nouvelle requête reformulée (expansée) Req_r est donc identifiée par la formule (IV.7).

$$Req_r = Req_0 \cup Req_{arbre} \quad \text{(IV.7)}$$

Où :

Req_0 : indique les termes de la requête originale.

Req_{arbre}: indique les termes des synsets (concepts) qui forment l'arbre sémantique à l'exception des concepts initiaux (requête originale) construit dans la section (*IV.2.3. Etape 3 : Construction de l'arbre sémantique).*

IV.2.5. Pondération les termes de la requête

Pour la pondération finale des termes de la requête, nous avons utilisé un poids (P) pour chaque terme dans la nouvelle requête reformulée Req_r selon le principe suivant :

- Si le terme *t* est un mot-clé d'origine, c'est-à-dire, il appartient à la requête originale (Req_0), alors le poids est égal à la valeur une ($P = 1$), car les mots-clés d'origine ont la meilleure indication du besoin de l'utilisateur ;
- Si le terme *t* est un mot-clé correspondant à un concept de l'arbre sémantique alors le poids *P* est calculé par la multiplication du poids du concept (nœud dans l'arbre) par la valeur 0,5 (c'est-à-dire $P=P_{arbre} * 0,5$. La valeur 0,5 est pour diminuer l'importance des termes à ajouter par rapport aux termes originaux de la requête).

Remarque : Si un terme appartient à deux catégories (Req_0 *et* Req_{arbre}) alors nous lui affectons le poids '1'.

IV.3. Test et expérimentation

Dans cette section nous présentons les expérimentations et les tests réalisés afin d'évaluer les performances de l'approche proposée, il faut rappeler que pour pouvoir évaluer une nouvelle technique, un nouvel algorithme ou une nouvelle approche intégrés dans un système de RI, nous devons utiliser une collection de test pour réaliser l'expérimentation et faire des comparaisons (*Chapitre I Section I.5. Collection de test des systèmes de RI*). Plus précisément, il faut donner un ensemble de requêtes au système de RI pour qu'il fasse la recherche dans l'index de la collection puis il sélectionne et il classe les documents retrouvés dans une liste ordonnée. L'évaluation n'est donc que le procédé de comparaison de ces résultats des documents récupérés par rapport aux documents réellement jugés préalablement pertinents par les experts.

Dans notre expérimentation nous suivons le protocole d'évaluation de TREC[4], en utilisant les mesures de précisions aux différents points suivants : à p@5, p@10, p@15, P@20, p@100 et la précision moyenne MAP (*Mean Average Precision*). Le procédés expérimental et les résultats sont décrits ci-dessous.

IV.3.1. Collection de test

Dans notre expérimentation, nous avons utilisé une collection constituée d'un corpus de textes journalistiques composée de 4.763 articles (documents)[5] collectés depuis l'agence de presse internationale de BBC News en arabe (Arabic BBC News[6]). Le corpus a été créé par (Saad & Ashour, 2010), dans le cadre du projet OSAC (Open Source Arabic Corpora –

[4] https://trec.nist.gov/
[5] https://sourceforge.net/projects/ar-text-mining/files/Arabic-Corpora/
[6] https://www.bbc.com/arabic

les corpus arabes open source-) présenté dans la 6eme édition de la conférence internationale EEECS'10. Le corpus est composé de 1.860.786 mots et 106.733 mots-clés après avoir supprimé les mots vides. Les thématiques ou les topics de ses documents couvrent plusieurs domaines comme montre le tableau IV.1. suivant.

Tableau IV.1. Corpus.

Topic (thème)		**Nombre de documents**
Nouvelles du Moyen-Orient	**أخبار الشرق الأوسط**	2.356
Nouvelles du monde	**أخبار العالم**	1.489
Économie et affaires	**اقتصاد وأعمال**	296
Sport	**رياضة**	219
Presse internationale	**صحافة عالمية**	49
Science & Technologie	**علوم وتكنولوجيا**	232
Arts & Culture	**فنون وثقافة**	122
Total		**4.763**

La collection inclut également une liste[7] de 43 requêtes sous divers topics (thèmes ou sujets), dont les jugements de pertinence ont été créés par nous-même à travers plusieurs projets PFE de master informatique. La Figure IV.6. suivante montre un exemple d'échantillons des requêtes (Q119, Q120, Q121).

- La requête *Q119* est : <التدخلات السياسية لباراك أوباما في العالم> <*L'ingérence politique de Barack Obama dans le monde*> avec 106 documents pertinents.
- La requête *Q120* est : <إنسحاب القوات الأمريكية من العراق > < *Retrait des forces américaines de l'Irak* > avec 35 documents pertinents.
- La requête *Q121* est : <سياسة دبي لتجاوز الأزمة الاقتصادية > < *La politique de Dubaï pour surmonter la crise économique* > avec 38 documents pertinents.

[7] https://sourceforge.net/projects/queries-for-arabic-osac-corpus/files/

```
- <query>
    <num>119</num>
    <title>التدخلات السياسية لباراك أوباما في العالم </title>
    <title_e> Barack Obama's political interventions in the world </title_e>
  - <keywords>
      <keyword>باراك أوباما</keyword>
      <keyword_e> Barack Obama </keyword_e>
      <keyword> سياسة أوباما </keyword>
      <keyword_e> Obama's politic </keyword_e>
      <keyword> العالم</keyword>
      <keyword_e> world </keyword_e>
    </keywords>
    <relev_doc>106</relev_doc>
  </query>
- <query>
    <num>120</num>
    <title>إنسحاب القوات الأمريكية من العراق </title>
    <title_e> the withdrawal of US forces from Iraq </title_e>
  - <keywords>
      <keyword>إنسحاب </keyword>
      <keyword_e> withdrawal </keyword_e>
      <keyword> القوات الأمريكية </keyword>
      <keyword_e> US forces </keyword_e>
      <keyword>العراق</keyword>
      <keyword_e> Iraq </keyword_e>
    </keywords>
    <relev_doc>35</relev_doc>
  </query>
- <query>
    <num>121</num>
    <title>سياسة دبي لتجاوز الأزمة الاقتصادية </title>
    <title_e> Dubai policy to overcome the economic crisis </title_e>
  - <keywords>
      <keyword>دبي </keyword>
      <keyword_e> Dubai </keyword_e>
      <keyword> سياسة دبي </keyword>
      <keyword_e> Dubai policy </keyword_e>
      <keyword>الأزمة الاقتصادية</keyword>
      <keyword_e> economic crisis </keyword_e>
    </keywords>
    <relev_doc>38</relev_doc>
  </query>
```

Figure IV.6. Exemple des requêtes (Q119, Q120, Q121).

En outre, pour adapter les valeurs des paramètres de l'étape2 de la section (*IV.2.2. Etape 2 : Extraction des concepts par PRF*), nous avons réalisé plusieurs séries de tests, de ce fait nous avons opté pour les valeurs optimales de ***K-top = 10*** documents et le seuil ***m = 0,6*** (*Les termes qui ont des valeurs de poids dépassant le seuil m sont sélectionnées comme termes candidats*).

IV.3.2. Procédure de test

Pour permettre d'expérimenter et d'évaluer l'approche proposée, nous avons réalisé trois séries de tests à travers l'utilisation du moteur de recherche Lucene[8] comme un système de RI:

1. Une série de test sans reformulation des requêtes : Ce test est utilisé comme l'évaluation de base (Baseline) afin de comparer les résultats.

2. Une série de test avec des concepts: Dans ce test, la requête est reformulée par l'ajout aveuglement de tous les termes des synsets (concepts pondérés) relatifs aux mots-clés de la requête originale après que les synsets ont été extraits depuis la ressource WordNet arabe et puis désambigüisés ainsi pondérés avec les techniques présentées dans la section (*IV.2.1.6. Extraction et désambiguïsation des concepts*).

   Exemple :

   - La requête ***Q116*** = *le mouvement rebelle au Soudan* (حركة التمرد في السودان), après son prétraitement elle génère la liste des mots-clés suivants : {Mouvement (حركة), rebelle(تمرد), Soudan (سودان)}, dont chaque mot-clé appartient à un concept (synset de WordNet arabe) :

     Synset **S1**: { حركة} (*motion.n.04*)

     Synset **S2**: { تمرد، جماح، استفزاز} (*rebelliousness, defiance, defiance.n.01*)

     Synset **S3**: { السُّودَان، جمهورية_السودان، سودان } (*sudan.n.01*)

   - La liste des mots-clés de la requête ***Q116*** reformulée devient {*Mouvement, rébellion, insurrection, provocation, Soudan, république du Soudan, Soudan*} {حركة، تمرد، جماح، استفزاز ، السُّودَان، جمهورية_السودان، سودان}

3. Une série de test avec la reformulation de la requête en utilisant l'approche proposée.

En effet, pour chaque série de test nous envoyons la liste des 43 requêtes à Lucene et par retour nous calculons les résultats sur les documents récupérés par deux mesures; la mesure par les précisions au rang *x* documents sélectionnés par le SRI (dont *x=5, 10, 15, 20 et 100*) et la mesure de la précision moyenne (MAP *Mean Average Precision*).

[8] https://lucene.apache.org/core/

Le tableau IV.2, la Figure IV.7 et la Figure IV.8 dressent une comparaison sur les résultats de ces trois séries de tests en fonction de deux mesures (la mesure de précision au rang X et la mesure du MAP)

Tableau IV.2. Valeurs des résultats expérimentaux des trois tests.

Précision à @x documents sélectionnés	**Test1 :** sans reformulation	**Test2** : reformulation de la requête par les concepts pondérés	**Test3 :** reformulation de la requête par notre approche
P@5	0,565	0,475	0,608
P@10	0,495	0,442	0,524
P@15	0,472	0,405	0,506
P@20	0,453	0,412	0,488
P@100	0,265	0,244	0,285
MAP	**0.248**	**0.232**	**0.273**

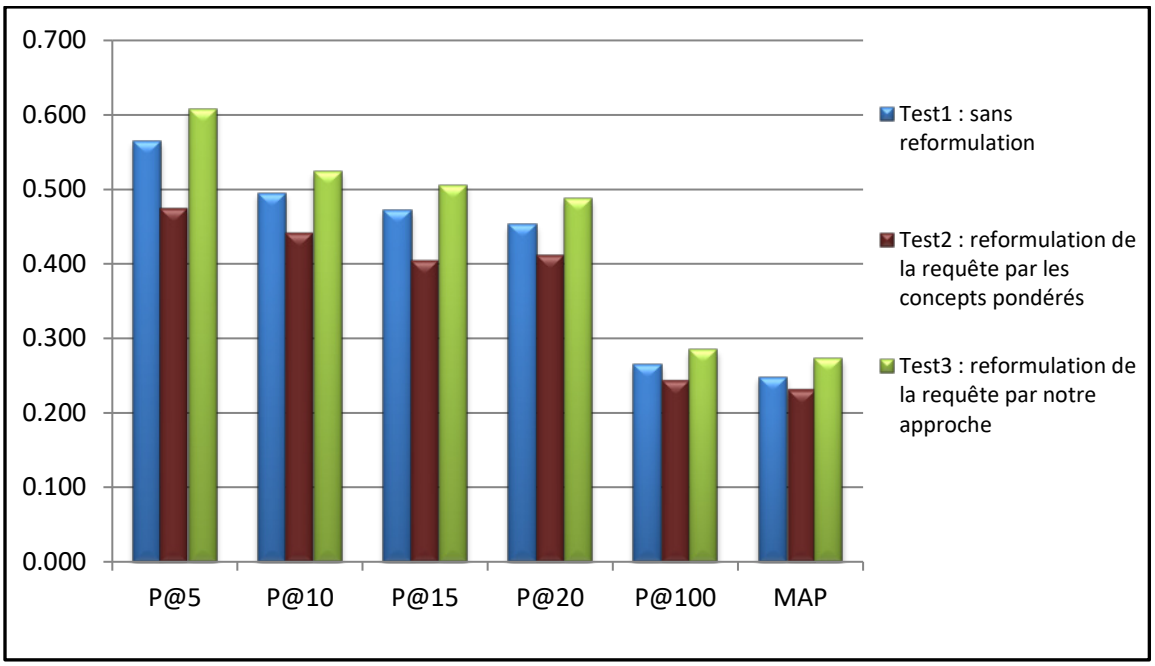

Figure IV.7. Représentation graphique des résultats.

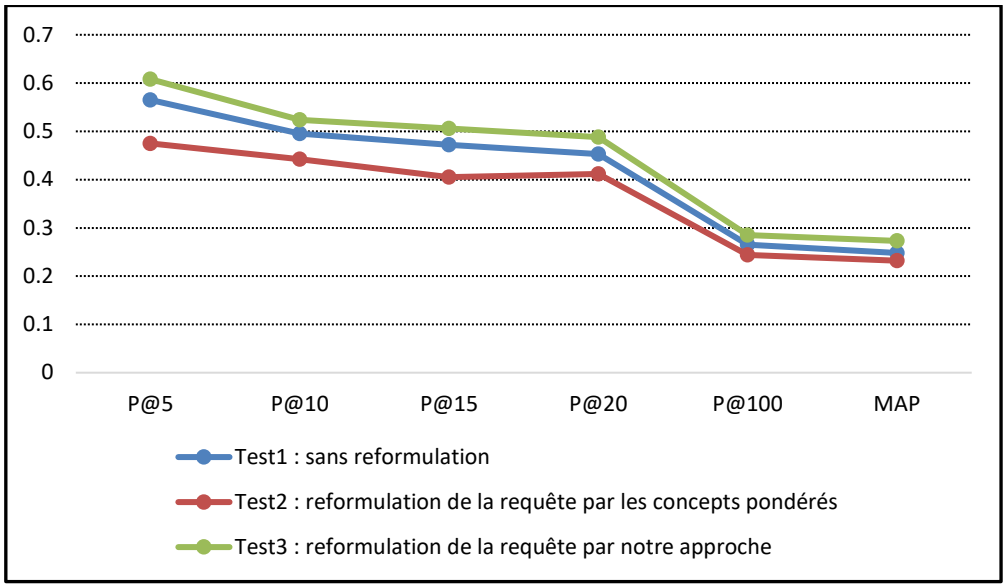

Figure IV.8. Performance des trois tests.

IV.3.3. Analyse des résultats

Pour analyser l'efficacité de l'approche, nous avons comparé les résultats obtenus par des pourcentages (%) de performance en deux rapports. Le premier rapport est la comparaison entre les résultats du ***Test2*** (par la reformulation des requêtes basée sur les concepts pondérés) par rapport aux résultats du ***Test1*** (Baseline sans reformulation des requêtes). Le deuxième rapport est la comparaison entre les résultats du ***Test3*** (par la reformulation des requêtes basée sur l'approche proposée) par rapport aux résultats du ***Test1*** (Baseline sans reformulation des requêtes). Le tableau IV.3. suivant mentionne les valeurs numériques de ces deux comparaisons.

Tableau IV.3. Valeurs des résultats expérimentaux.

Précision	Test2 % Test1	Test3 % Test1
P@5	-15,93%	7,61%
P@10	-10,71%	5,86%
P@15	-14,19%	7,20%
P@20	-9,05%	7,73%
P@100	-7,92%	7,55%
MAP	-6,45%	**10,08%**

Selon les valeurs résultats des expérimentations présentées dans le tableau IV.3 nous concluons ce qui suit :

- Une mauvaise performance lorsque nous utilisons la reformulation (expansion) des requêtes par uniquement des concepts pondérés à l'ordre de : *-15,93 %, -10,71 %, -14,19 %, -9,05 %, -7,92 %* et *-6,45 %* pour respectivement les précisions aux rangs des points, *p@5, p@10, p@15, P@20* et *p@100*, ainsi pour la *MAP*. Apres investigation nous avons constaté que ces mauvaises performances sont principalement dues à la faiblesse de la couverture de la ressource WordNet arabe (AWN) par rapport à la langue, étant donné que seulement *68%* des mots-clés des requêtes se trouvaient dans l'AWN où ils sont conceptualisés.
- La reformulation basée sur l'arbre sémantique a donné une meilleure performance par rapport à Baseline (sans reformulation des requêtes) avec des valeurs stables sur tous les points de précisions à l'ordre de *7,61 %, 5,86 %, 7,20 %, 7,73 %, 7,55 %* et *10,08 %* pour *p@5, p@10, p@ 15, P@20, p@100* et *MAP* respectivement. L'expérimentation a permis donc de montrer que la reformulation des requêtes à travers l'arbre sémantique construit suivant l'approche proposée converge davantage pour décrire le besoin d'information de l'utilisateur, ce qui conclut que cette technique améliore les performances des systèmes de la recherche d'information arabe.

IV.4. Discussion sur l'approche proposée

Suite aux résultats expérimentaux parvenus précédemment, la méthode proposée dans ce projet de thèse démontre une bonne amélioration de performance de la précision moyenne (MAP *Mean Average Precision)*, elle est d'environ de 10% d'amélioration du système de RI lors de l'utilisation de l'arbre sémantique pour la reformulation (expansion) des requêtes afin de capturer le besoin de l'utilisateur. Nous pensons que ces performances indiquent que cette méthode est encourageante, puisque les techniques basées sur les concepts utilisant le WordNet arabe et la méthode de pseudo-réinjection de la pertinence (Pseudo Relevance feedback PRF) présentées dans le travail de (Abderrahim, 2014) ont seulement amélioré le SRI d'environ 4%.

En revanche, la méthode qui utilise le WordNet arabe et les règles d'association basées sur des relations afin de sélectionner les synonymes appropriés des termes des requêtes, présentée par les chercheurs (Abbache, Meziane, Belalem, & Belkredim, 2018) dans leur travail a montré une augmentation des performances d'environ de 13% de MAP, mais les résultats des tests n'étaient que sur un ensemble réduit des mots-clés et elle est évaluée sur une sous-collection de RI.

Par ailleurs, la technique proposée par (Karisani, Rahgozar, & Oroumchian, 2016) qui se basait sur l'analyse locale, pour identifier et re-pondérer les termes informatifs des requêtes, n'a amélioré les performances du système de RI que d'environ de 7% de MAP par rapport aux méthodes traditionnelles de re-pondération des termes.

En conclusion, nous affirmons à travers l'actuel travail qu'il existe une bonne solution pour représenter les mots-clés de la requête ou le besoin de l'utilisateur par une hiérarchie sémantique. Également, nous avons montré que l'utilisation aveugle de la ressource du WordNet arabe afin de reformuler des requêtes est limitée et elle n'améliore pas les performances des systèmes de RI, puisque cette ressource ne couvre pas assez la langue arabe et même elle contient de nombreuses erreurs ce qui réaffirme aussi le témoignage des chercheurs pour le Princeton WordNet pour l'anglais (McCrae & Prangnawarat, 2016).

IV.5. Conclusion

Au long de ce chapitre, nous nous sommes intéressés à proposer et d'expérimenter une nouvelle technique de reformulation des requêtes afin d'améliorer les performances du système de recherche d'information pour les documents textuels arabes.

Bien que les méthodes et les techniques de reformulation des requêtes sont largement étudiées et utilisées pour améliorer les performances des systèmes de RI surtout pour les autres langues hors l'arabe, à travers notre contribution, nous avons proposé de représenter la requête à travers un arbre sémantique. Cet arbre est généré à partir des concepts relatifs aux mots-clés de la requête originale et des concepts des termes relatifs à la technique de PRF ainsi que de leurs extensions conceptuelles correspondantes par des relations de synonymie, d'hyponymie et d'hyperonymie. À cet effet, la ressource du WordNet arabe a été utilisée à la fois pour la désambiguïsation des concepts et la hiérarchisation de l'arbre. Enfin, le procédé de la reformulation (expansion) est donc opéré par l'ajout directe des termes pondérés extraits de cet arbre à la requête originale.

Les résultats expérimentaux utilisant la métrique de précision au points (@5, @10, @15, @20 et @100) et la précision moyenne (MAP) ont montré; une diminution des performances lorsque les requêtes sont aveuglément étendues par des concepts, ceci est principalement dû à la faible couverture sémantique de la langue arabe par la ressource du WordNet arabe, cependant, une amélioration positive dans les performances du système de RI autour de 10% de MAP après application de notre approche proposée.

Encore une fois, ces résultats donnent des pistes intéressantes pour des futurs travaux sur la langue arabe utilisant des arbres sémantiques pour capturer les besoins d'informations des utilisateurs.

Dans le chapitre suivant, nous allons aborder la problématique de création et d'évaluation d'une nouvelle collection de test pour la recherche d'information arabe.

Chapitre V. Création d'une collection de test pour des systèmes de RI arabe basée sur la stratégie de Pooling et l'apprentissage automatique

V.1. Introduction

Nous avons montré dans le chapitre III, qu'ils existent très peu de collections avec des jugements de pertinence pour tester et évaluer des systèmes de recherche d'information ou des moteurs de recherche pour la langue arabe, et encore, la plupart d'elles ne sont ni gratuites ni disponibles en ligne par une licence d'accès libre (open source).

De plus, la construction de telles collections de test est un processus complexe et coûteux mais il reste indispensable afin de déterminer les meilleures techniques et pour évaluer les systèmes de RI, les moteurs de recherche, les nouveaux algorithmes, les requêtes, les corpus ou les métriques. Les collections de test peuvent être aussi utilisées dans d'autres domaines tels que le filtrage, la classification ou la catégorisation (clustering) des documents.

L'objectif de ce chapitre est de présenter notre deuxième contribution par la proposition d'une méthode pour la création d'une nouvelle collection de test de RI pour la langue arabe.

Contrairement aux études précédentes, notre méthode propose une combinaison de deux techniques, la stratégie de Pooling et l'algorithme Naïve-Bayes d'apprentissage automatique. Le travail commence alors par la construction d'un corpus à travers la collecte des documents, ensuite la création d'une liste des requêtes avec leurs jugements de pertinence en se basant sur la stratégie de Pooling. Puis, l'application de l'algorithme Naïve-Bayes afin de tester et d'évaluer ces jugements de pertinence. Enfin, le modèle word2vec est aussi utilisé pour enrichir sémantiquement la base documentaire afin d'améliorer les performances du classificateur Bayésien.

Le présent chapitre est organisé comme suit ; tout d'abord, dans la section V.2, nous présentons un aperçu par un schéma sur la méthode proposée. La section V.3 détaille la construction de la nouvelle collection ainsi que ses caractéristiques. Dans la section V.4, nous expliquons comment la stratégie de Pooling est appliquée afin de créer les jugements de pertinence. Ces jugements sont évalués aussi par le classificateur Bayésien dans la section V.5. Nous terminons le chapitre par une discussion sur les résultats expérimentaux dans la section V.6 et une conclusion dans la section V.7.

V.2. Méthode proposée

Nous nous intéressons ici à proposer une méthode pour la construction d'une nouvelle collection de test de RI, cette dernière contient principalement des documents en arabe mais aussi nous avons inclus des documents anglais afin de permettre à la collection d'être utilisée pour tester des requêtes bilingues ou même plus dans le domaine de la traduction automatique.

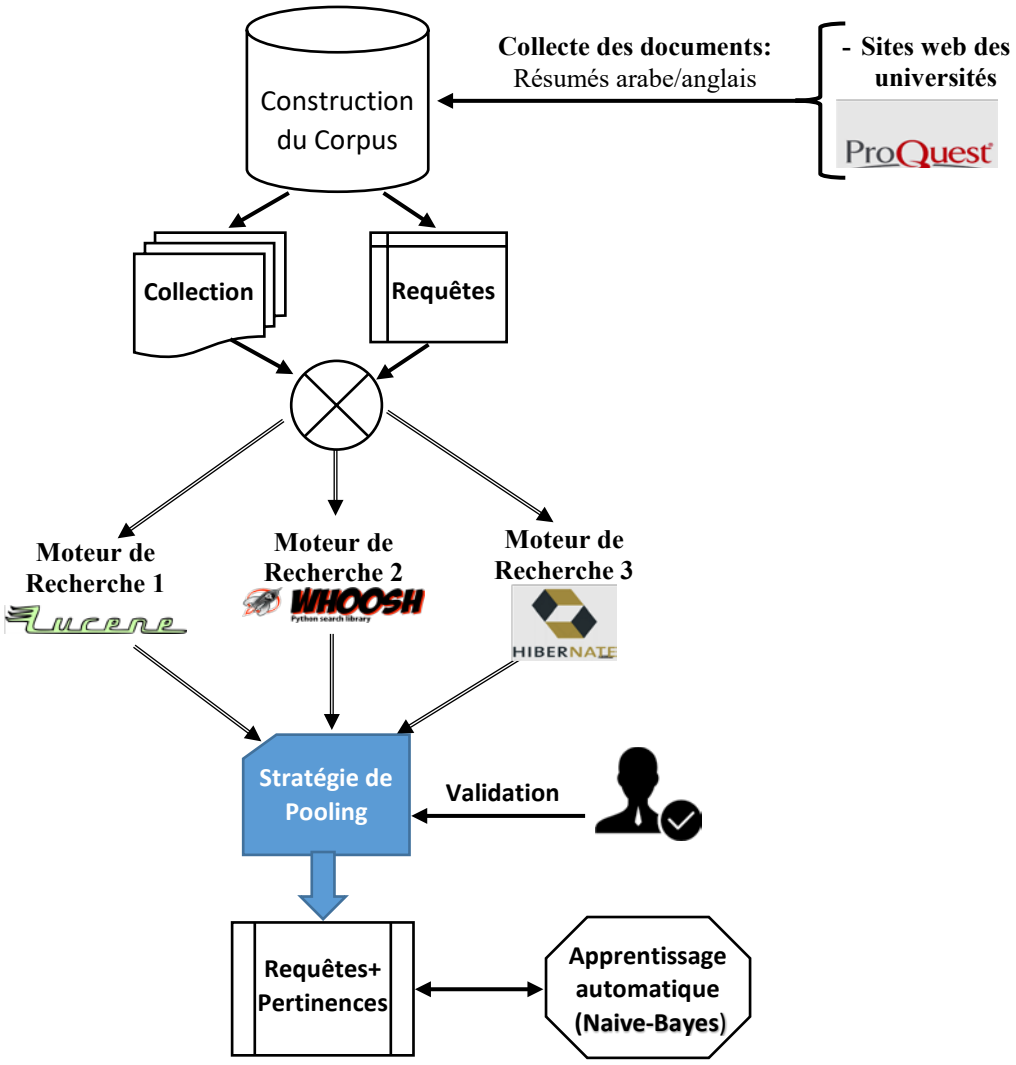

Figure V.1. Diagramme de la méthode proposée.

La Figure V.1 résume la méthode proposée en trois principales étapes. Premièrement, la création et la construction du corpus parallèle (arabe/anglais) à partir du Web ainsi que la constitution de la liste des requêtes.

Dans la deuxième étape, nous utilisons le principe de la technique de *Pooling*, tel qu'il est utilisé dans les travaux de recherche de (Tonon, Demartini, & Cudré-Mauroux, 2015; Voorhees & Harman, 2005). Cette technique sert alors à créer une collection de documents par mettre ensemble des N résultats de sélection des documents retournés par X moteurs de recherche ou de systèmes de RI, cet ensemble de documents est appelé par un 'Pool'. Ensuite, des humains jugent manuellement la pertinence ou la correspondance de chaque document de ce 'Pool à une requête donnée, et lorsqu'un document se trouve en dehors du 'Pool', il est automatiquement considéré comme non pertinent. Dans notre contribution, nous utilisons trois moteurs de recherche afin de créer ce Pool, puis, en s'appuyant sur leurs résultats de sélections des documents, nous calculons automatiquement le degré de la pertinence correspondants aux requêtes de chaque document.

Pour la dernière étape, nous appliquons l'algorithme Naïve-Bayes d'apprentissage automatique pour valider les résultats des pertinences obtenus. Pareillement, l'algorithme est aussi testé une deuxième fois sur la collection après son enrichissement sémantique par le modèle Word2vec.

V.3. Construction de la collection

V.3.1. Collecte des documents

Afin de construire le corpus, nous avons collecté 632 documents textuels bilingues (arabe/anglais), ces documents sont formés par les résumés des rapports de thèses de doctorat et de magister, dont chaque texte arabe collecté, son texte correspondant en anglais est aussi récupéré. La taille moyenne des documents est de 203 mots pour le texte arabe et 218 mots pour le texte anglais. Le corpus est principalement (90%) téléchargé depuis la bibliothèque en ligne *ProQuest library* [1] , le reste des 10% des documents sont téléchargés depuis les sites web des universités. Bien évidement nous avons utilisé les thèses de doctorat et de magister de la région du monde arabe (Algérie, Maroc, Arabie Saoudite et le Qatar) pour avoir les documents parallèles arabe-anglais. Nous avons choisi les thèses comme source documentaire pour avoir une fiabilité de traduction des textes dans les deux sens (arabe/anglais).

V.3.2. Caractéristiques de la collection

Au moment de la collecte du corpus, nous avons récupéré aussi pour chaque document les méta-informations principales suivantes : le titre du document en arabe et en anglais, le nom de l'université en arabe et en anglais, l'année de publication de la thèse, le topic (sujet) en arabe et en anglais, le sous-topic en arabe et en anglais et les mots-clés en arabe et en anglais. La Figure V.2 montre un aperçu sur ces méta-informations ainsi le tableau V.1 illustre plus de détails sur les caractéristiques de ces informations.

[1] https://www.proquest.com/

Concernant les informations manquantes dans certains documents, par exemple les mots-clés absents en arabe ou en anglais, nous les avons complété par la traduction automatique via la librairie « Google Translate », également après la vérification et la validation manuelle.

Id-Doc	Arabic_Title	English_Title	Ar-University_Name	Eng-University_Name	Year	Ar_Topic	Eng_Topic	Ar-Sub_Topic	Eng-Sub_Topic	Arabic Keywords
D001	دراسة وتنفيذ الخدمة التجريبية لرايلر توب	Study and implementation of pilot service	الجامعة الدولية بالدار البيض	International University	2009	تكنولوجيا	Technology	هندسة الاتصالات	Telecommuni	ة؛ المحاسبة المكالمة؛ الأمن
D002	الأخطاء التركيبية لدى متعلمي اللغة العربية طلبة	Syntactic errors among Arabic language	جامعة حمد بن خليفة ، قطر	Hamad Bin Khalifa Uni	2016	العلوم الادبية	Literary scie	العلوم الإسلامية	Islamic scien	ربية؛ طلبة؛ قطر
D003	الاصلاح الأسري في ضوء الشريعة و النظريات ا	Family reform in the light of Sharia and	جامعة حمد بن خليفة ، قطر	Hamad Bin Khalifa Uni	2017	العلوم الادبية	Literary scie	العلوم الإسلامية	Islamic scien	، الأسري؛ الشريعة؛ النظريات التفـ
D004	دراسة مقارنة بين فرضيات دولة العقيدة والأمة	A comparative study between the assum	جامعة حمد بن خليفة ، قطر	Hamad Bin Khalifa Uni	2017	العلوم الادبية	Literary scie	العلوم الإسلامية	Islamic scien	، دولة العقيدة؛ الأمة؛ الفرضيات ال
D005	توصيف الأجسام المضادة وحيدة النسيلة الفأرية	Characterization of Mouse Monoclonal A	جامعة حمد بن خليفة ، قطر	Hamad Bin Khalifa Uni	2017	العلوم الطبيع	Natural Scie	كيمياء طبيعية	Natural Chem	المضادة وحيدة النسيلة الفأرية؛
D006	المنهاج القرآني في تقويم العاطفة تجاه الأهل وا	The Qur'anic curriculum in evaluating the	جامعة حمد بن خليفة ، قطر	Hamad Bin Khalifa Uni	2017	العلوم الادبية	Literary scie	العلوم الإسلامية	Islamic scien	القرآني؛ العاطفة؛ الأهل؛ العشيرة
D007	التأثير الإسلامي في فكر العالم اليهودي سعيد بن	The Islamic Influence in the Thought of t	جامعة حمد بن خليفة ، قطر	Hamad Bin Khalifa Uni	2017	العلوم الادبية	Literary scie	العلوم الإسلامية	Islamic scien	سلامي؛ العالم اليهودي؛ سعيد بن
D008	تقدير وتصفية التوافقيات	estimation and filtering of harmonics	جامعة الملك فهد للبترول و	King Fahd University o	1997	تكنولوجيا	Technology	هندسة كهربائية	Electrical Eng	تصفية؛ التوافقيات؛ كهربائي
D009	دراسة EPR لمجمعات الفاناديل مع قواعد شيف	EPR study of vanadyl complexes with Sc	جامعة الملك فهد للبترول و	King Fahd University o	1992	العلوم الدقيقة	Exact Scienc	الكيمياء	Chemistry	عد ؛ الهيدرازين؛ كربوديثيوات
D010	التحليل الكيميائي لبعض عقارات البنسلين والسيفـ	Chemical analysis of some penicillin and	جامعة الملك فهد للبترول و	King Fahd University o	1991	العلوم الدقيقة	Exact Scienc	الكيمياء	Chemistry	الكيميائي؛ البنسلين؛ السيفالوسبو
D011	دراسة السياسة العامة لمكافحة التلوث النفطي الـ	Study the general policy to combat mari	جامعة حمد بن خليفة ، قطر	Hamad Bin Khalifa Uni	2017	تكنولوجيا	Technology	هندسة البترول	Petrol Engine	التلوث النفطي؛ النفط البحري؛ الت
D012	الدولة الإسلامية تساؤلات الضرورة والإستحالة	Islamic state questions of necessity and	جامعة حمد بن خليفة ، قطر	Hamad Bin Khalifa Uni	2017	العلوم الادبية	Literary scie	العلوم الإسلامية	Islamic scien	لمسلمين؛ تصنيف الأديان؛ تاريخ ا
D013	جهود علماء المسلمين في تصنيف الأديان في ض	The efforts of Muslim scholars in the clas	جامعة حمد بن خليفة ، قطر	Hamad Bin Khalifa Uni	2017	العلوم الادبية	Literary scie	العلوم الإسلامية	Islamic scien	وقف؛الفقه الإسلامي؛ الفوائد البنكـ
D014	تطوير محرك بحث وفهرسة مستندات قرآنية	Development of a search and indexing e	المدرسة الوطنية لعلوم الحـ	National School of Cor	2010	تكنولوجيا	Technology	علوم الكومبيوتر	Computer Sc	محرك البحث ؛ الفهرسة ؛ الاستـ
D015	niqueAuthentic تصور وتنفيذ محرر النص	Conception and realization of a text edito	المدرسة الوطنية لعلوم الحـ	National School of Cor	2015	تكنولوجيا	Technology	علوم الكومبيوتر	Computer Sc	حث النص ؛ بروتوكولات الأمان ؛
D016	الخصائص المحلية لاحد الرواسم الهرمونية من ن	Local Properties of Some Polynomial Cl	جامعة الملك فهد للبترول و	King Fahd University o	1995	العلوم الدقيقة	Exact Scienc	الرياضيات	Mathematics	ت؛ البحث؛ التقارب ؛ كثيرات الحدو
D017	تدخلات محفزة إيدولوجيا في الترجمة العربية: تد	Advocacy-driven interventions in Arabic	جامعة حمد بن خليفة ، قطر	Hamad Bin Khalifa Uni	2017	العلوم الادبية	Literary scie	العلوم الاجتماعية	Social Scienc	وجيا ؛ نظرية التقييم ؛ مقارنة الن
D018	الإعلان الإسلامي بشأن تغير المناخ: حل للاحترار	Islamic Declaration on Climate Change:	جامعة حمد بن خليفة ، قطر	Hamad Bin Khalifa Uni	2017	العلوم الادبية	Literary scie	العلوم الإسلامية	Islamic scien	لدالة؛ القرآن الكريم؛ دراسة دلالية
D019	استثمار أموال الوقف في الفقه الإسلامي والتجرا	Investing Waqf funds in Islamic jurispru	جامعة حمد بن خليفة ، قطر	Hamad Bin Khalifa Uni	2017	العلوم الادبية	Literary scie	العلوم الإسلامية	Islamic scien	ستثمار؛ صناديق الوقف؛ الفقه الإ
D020	التربية الإسلامية في أواخر اسطنبول العثمانية:	Islamic Education in Late Ottoman Istan	جامعة حمد بن خليفة ، قطر	Hamad Bin Khalifa Uni	2016	العلوم الادبية	Literary scie	العلوم الإسلامية	Islamic scien	(908 سلامية ؛ اسطنبول العثمانية

Figure V.2. Aperçu sur la méta-information du corpus.

Tableau V.1. Caractéristiques du corpus.

Information	Type	Valeur Min	Valeur Max	Valeur Moyenne
Id-Doc	Chaîne de caractères	D001	D0632	
Titre arabe	Chaîne de caractères	4 mots	23 mots	~9.3 mots
Titre anglais	Chaîne de caractères	4 mots	19 mots	~8.6 mots
Nom de l'université en arabe	Chaîne de caractères	5 mots	8 mots	
Nom de l'université en anglais	Chaîne de caractères	5 mots	9 mots	
Année de publication de la thèse	Date (année)	1982	2019	
Topic (sujet) en arabe	Chaîne de caractères	T1	T6	
Topic (sujet) en anglais	Chaîne de caractères	T1	T6	
Sous-Topic en arabe	Chaîne de caractères	ST01	ST19	
Sous-Topic en anglais	Chaîne de caractères	ST01	ST19	
mots-clés en arabes	Chaîne de caractères	3 mots	14 mots	~7.3 mots
mots-clés en anglais	Chaîne de caractères	3 mots	14 mots	~7.3 mots

Le tableau IV.2 résume aussi la base documentaire selon la répartition des années de publications des thèses.

Tableau V.2. Répartition des documents selon les années de publications.

Année de publication	Avant 1990	1990-1999	2000-2009	2010-2019	Total
Nombre de documents	23	180	194	235	**632**
En %	3.64%	28.48%	30.70%	37.18%	**100%**

Nous avons montré dans le chapitre I que la base documentaire des collections de test doit être suffisamment diversifiées par les besoins d'information des utilisateurs, afin de pouvoir tester les systèmes de RI sous plusieurs paramètres, pour cette raison nous avons sélectionné divers domaines de recherche des thèses pour créer un nombre suffisant des topics et des sous-topics. Le tableau V.3 suivant détaille les topics et les sous-topics choisis en trois langue (français, arabe et anglais) ainsi que le nombre de documents pour chaque topic.

Tableau V.3. Topics et sous-topics de la collection.

Id Topic	Topic en français	Topic en arabe	Topic en anglais	Nbre de Docs	Id sous-topic	Sous topic en français	Sous topic en arabe	Sous topic en anglais
T1	Sciences naturelles	علوم طبيعية	Natural Sciences	33 Docs	**ST11**	Chimie Naturelle	الكيمياء الطبيعية	Natural Chemistry
					ST12	Génie de l'environne-ment	الهندسة البيئية	Environmen-tal Engineering
					ST13	Géologie	جيولوجيا	Geology
T2	Technologie	تكنولوجيا	Technology	325 Docs	**ST21**	Ingénierie pétrolière	هندسة البترول	Petrol Engineering
					ST22	Ingénierie mécanique	الهندسة الميكانيكية	Mechanical Engineering
					ST23	Ingénierie électrique	الهندسة الكهربائية	Electrical Engineering
					ST24	Informatique	علوم الكومبيوتر	Computer Sciences
					ST25	Ingénierie aéronautique	هندسة الطيران	Aviation Engineering
					ST26	Télé-communication	هندسة الاتصالات	Telecommuni-cation Engineering

T3	Sciences exactes	العلوم الدقيقة	Exact Sciences	107 Docs	**ST31**	Mathématiques	الرياضيات	Mathematics
					ST32	Physique	الفيزياء	Physics
					ST33	Chimie	الكيمياء	Chemistry
T4	Ingénierie de la ville	هندسة المدينة	City Engineering	115 Docs	**ST41**	Ingénierie des transports	هندسة المواصلات	Transportation Engineering
					ST42	Génie civile	هندسة مدنية	Civil Engineering
					ST43	Gestion administrative et construction	هندسة الإدارة والبناء	Management and Construction
T5	Économie	الاقتصاد	Economy	19 Doc	**ST51**	Économie	الاقتصاد	Economy
T6	Sciences littéraires	العلوم الادبية	Literary sciences	33 Docs	**ST61**	Sciences islamiques	العلوم الاسلامية	Islamic Sciences
					ST62	Sciences sociales	العلوم الاجتماعية	Social Sciences
					ST63	Langue arabe	الغة العربية	Arabic Language

V.3.3. Création de la liste des requêtes

Il est nécessaire de créer une liste de requêtes sous plusieurs topics (sujets), de vocabulaire plus ou moins lié au domaine de la collection afin d'imiter étroitement les requêtes réelles des utilisateurs. Aussi, la liste doit inclure des requêtes de différentes tailles, courtes et moyenne, voire aussi des requêtes qui contiennent des entités spécifiques (par exemple, des personnes ou des lieux).

Dans le procédé de la création de cette liste de requêtes, nous suivons les trois principales phases suivantes :

Phase 01 :

Tout d'abord, nous initialisons une liste des requêtes candidates à partir des titres des documents (thèses) et les listes des mots-clés écris seulement en arabe, sous le critère d'avoir une taille limitée par le nombre de mots composants qui ne dépasse pas un seuil (nous avons déterminé expérimentalement ce seuil à 4 mots), nous limitons ici la taille des requêtes pour avoir des requêtes non ambigües, c'est-à-dire chaque requête-candidate ne représente qu'un seul besoin d'information. Techniquement, nous appliquons la méthode de n-gramme dont nous varions le n de 1 à 4.

Le tableau V.4 résume un exemple de sous-chaines générées par la technique de n-gramme pour la chaine suivante Ch1.

Ch1= « اهداف ترشيد إستهلاك الطاقة الكهربائية » (*Objectifs de rationalisation de la consommation d'énergie électrique*).

Tableau V.4. Exemples de sous-chaines générées par n-gramme.

n-gramme	Texte en arabe	Texte en français
Texte original	اهداف ترشيد إستهلاك الطاقة الكهربائية	Objectifs de rationalisation de la consommation d'énergie électrique
Uni-gramme	اهداف	Objectifs
Uni-gramme	ترشيد	Rationalisation
Uni-gramme	إستهلاك	Consommation
Uni-gramme	الطاقة	Energie
Uni-gramme	الكهربائية	Electrique
Bi-gramme	اهداف ترشيد	Objectifs de rationalisation
Bi-gramme	ترشيد إستهلاك	Rationaliser la consommation
Bi-gramme	إستهلاك الطاقة	Consommation d'énergie
Bi-gramme	الطاقة الكهربائية	L'énergie électrique
Tri-gramme	اهداف ترشيد إستهلاك	Objectifs de rationalisation de la consommation
Tri-gramme	ترشيد إستهلاك الطاقة	Rationalisation de la consommation d'énergie
Tri-gramme	إستهلاك الطاقة الكهربائية	Consommation d'énergie électrique
Quadri-gramme	اهداف ترشيد إستهلاك الطاقة	Objectifs de rationalisation la consommation d'énergie
Quadri-gramme	ترشيد إستهلاك الطاقة الكهربائية	Rationalisation de la consommation d'énergie électrique

Phase 02 :

En se basant sur le principe que la fréquence d'apparition d'une requête-candidate à un certain nombre peut indiquer que cette requête-candidate corresponde à un besoin d'utilisateur. De ce fait, nous calculons les fréquences de requête-candidates dans toute la base documentaire, et celles qui dépassent le seuil maximum (*Fq_max*) sont éliminées puisque les requêtes- candidates fréquentes n'expriment plus les besoins précis des utilisateurs, elles deviennent comme des expressions vides de sens. Aussi nous éliminons celles dont la fréquence est moins du seuil minimum (*Fq_min*), puisqu'aussi les requêtes rares perturbent le calcul d'évaluation du SRI. En conséquence, nous ne préservons que les

requêtes-candidates qui ont une fréquence entre ces deux bornes pour la phase suivante. La formule (V.1) suivante résume le principe de cette phase.

Après une série de tests, nous avons défini expérimentalement le *Fq_min* =15 et le *Fq_max*=100.

$$Fq\_min <= Freq(Q\_c_i) <= Fq\_max \quad \text{(V.1)}$$

Où : Q_c_i indique la requête candidate.

Phase 03 :

Dans la phase3, nous validons manuellement le résultat précèdent des requêtes candidates, pour cela nous avons sollicité trois personnes natives parlant l'arabe (étudiants en master2 et doctorants) afin de vérifier toute la liste et de ne conserver notamment que les requêtes significatives et représentatives à des besoins réels d'information. Après cette vérification et validation nous avons obtenu une liste de 165 requêtes. Nous avons opté pour ce nombre de requête (165) en s'appuyant sur la conclusion du travail de (Carterette, Pavlu, Kanoulas, Aslam, & Allan, 2008) qui a montré qu'avec un nombre de topics réduits mais avec un grand nombre de requêtes conduit à une évaluation plus fiable. Le tableau V.5 résume un échantillon de cette liste.

Tableau V.5. Échantillon de la liste des requêtes validées.

Id requête	Requête en français	Requête en arabe	Requête en anglais
Q001	Sécurité des informations	الأمن المعلوماتي	Information security
Q002	Connexion au réseau intelligent	اتصال الشبكة الذكية	Smart Grid connection
Q003	Performances des systèmes de communication	أداء انظمة الاتصالات	Performance of communication systems
Q004	Investissement des fonds Waqf	استثمار أموال الوقف	Investment of Waqf funds
Q005	utilisation d'enzymes	استعمال الانزيمات	use of enzymes
Q006	travail d'entretien	أعمال الصيانة	Maintenance work
Q007	Oxyde d'aluminium	اكسيد الألمنيوم	Aluminium oxide
Q008	Impact islamique	الأثر الإسلامي	Islamic Impact
Q009	Antiviraux	الأجسام المضادة للفيروسات	Antivirals
Q010	Durabilité sociale	الاستدامة الاجتماعية	Social sustainability
.	…	…	…
.	…	...	…
Q163	Utilisation de l'eau	استخدام المياه	Water use
Q164	Système énergétique	نظام الطاقة	Energy system
Q165	Oxyde et oxydation	الأكسدة	Oxide and Oxidation

La liste inclut également des requêtes de différentes tailles, la longueur de ces tailles varie entre un (1) au quatre (4) mots en arabe, le tableau V.6. illustre ainsi le nombre des requêtes selon leurs tailles.

Tableau V.6. Nombre des requêtes selon leurs tailles.

Longueur de la requête	Nombre de requêtes
Un Mot	22
Deux mots	105
Trois Mots	31
Quatre Mots	7
Total	**165**

V.4. Stratégie de Pooling

Dans le premier chapitre, Nous avons présenté lors de la construction d'une collection de test de RI que, parmi les techniques de création des jugements de pertinence liées à des requêtes est la technique appelée stratégie de Pooling.

Egalement, nous avons vu que cette technique a été adoptée par TREC[2] dès ses débuts. En fait, son principe sert à rassembler les N premiers documents résultats des différents moteurs de recherche (ou des systèmes de RI) testés pour chaque requête et à regrouper tous les résultats en une seule liste de résultat (appelée Pool), par la suite, cette liste est donnée aux évaluateurs (experts humains) pour la validation afin de créer finalement les jugements de pertinence.

Dans notre approche, nous avons adopté cette technique en se basant aussi sur le principe présenté par (Radlinski & Craswell, 2010), dont les chercheurs ont montré qu'il existe une étroite corrélation entre les jugements d'experts et le résultat de la méthode d'entrelacement basé sur la stratégie de Pooling, c'est-à-dire que lorsqu'une requête est jugée par un expert vaut presque le résultat de cette méthode.

Dans ce cadre, nous avons appliqué cette méthode d'entrelacement basée sur la stratégie de Pooling afin de calculer la pertinence des documents liés aux requêtes, dans lequel, le résultat du Pool est créé par l'exécution de trois moteurs de recherche, dont deux parmi eux fonctionnent hors ligne (Lucene[3] et Whoosh[4]) et le troisième fonctionne en ligne

[2] Text REtrieval Conference https://trec.nist.gov/
[3] https://lucene.apache.org/core/6_5_1/index.html
[4] https://pypi.org/project/Whoosh/

(Hibernate[5]), ces trois moteurs de recherche sont exécutés pour chaque requête en deux langues (arabe et anglais) ce qui crée six résultats différents.

Pour rappel :

- **Lucene** est un package Java qui fournit des fonctions d'indexation et de sélection des documents; il est gratuit et appliqué pour évaluer de nouvelles approches et d'algorithmes de recherche. Aussi, ce moteur de recherche est très populaire dans le web avec un résultat de temps d'exécution très rapide, il est même utilisé par de nombreuses plateformes[6] comme *LinkedIn* et *Twitter*.

- **Whoosh** est aussi une bibliothèque de recherche open source implémentée en Python, elle est rapide et elle réalise une bonne indexation des documents, elle permet ainsi la gestion simple de l'index des documents, des requêtes et du classement (Ranking des documents), chaque partie de ses fonctionnalités peut être remplacée ou étendue pour répondre exactement aux différents besoins des programmeurs.

- **Hibernate** est une plateforme sous Java sponsorisée par RedHat Apache, elle offre plusieurs fonctionnalités sous le projet « recherche ouverte des données » (opensearch[7]), elle permet également l'indexation des bases textuelles et la sélection des documents.

V.4.1. Création des Pools

Nous exécutons les trois moteurs de recherche (Lucene, Whoosh et Hibernate) pour chaque requête en deux langues (arabe et anglais). Plus précisément, pour la requête en arabe le moteur de recherche cherche dans la base documentaire des textes arabes et il récupère un pool, c'est idem pour la requête en anglais, puisque pour chaque document arabe il existe son équivalent en anglais, alors le moteur de recherche fait la recherche dans la base documentaire en anglais et il récupère également un deuxième Pool.

En conséquence, les trois moteurs de recherche retournent au total six résultats de Pools différents. Par la suite, nous utilisons l'algorithme d'entrelacement équilibré (*Balanced interleaving* illustré par la Figure V.3) proposé par (Chapelle, Joachims, Radlinski, & Yue, 2012), cet algorithme permet de regrouper les six résultats de Pools retournés en un seul classement intégré comme c'est présentée dans la Figure V.4.

[5] http://hibernate.org/
[6] https://cwiki.apache.org/confluence/display/LUCENE/PoweredBy
[7] https://www.opensearch.org/

Algorithme 01 : Entrelacement équilibré (*Balanced interleaving*)

Début
Lire (Liste_A (a_1, a_2, ...) **et** Liste_B (b_1, b_2, ...))
$I \leftarrow [\,]$
$K_a \leftarrow 1$
$K_b \leftarrow 1$
$AFirst \leftarrow RandomBit(0,1)$
Tant que $(K_a \leq |A|$ ***et*** $K_b \leq |B|)$ **Faire** *{Tant que non de fin de liste_A ou de liste_B}*
    Début
        Si $(K_a < K_b)$ ***ou*** $((K_a = K_b)$ ***et*** $(AFirst = 1))$ **Alors**
            Si $(A[K_a] \notin I)$ Alors $I \leftarrow I + A[K_a]$ *{Ajouter élément de la liste_A à la liste I}*
            $K_a{+}{+}$
        Sinon
            Si $(B[K_b] \notin I)$ Alors $I \leftarrow I + B[K_b]$ *{Ajouter élément de la liste_B à la liste I}*
            $K_b{+}{+}$
        Fin_Si
    Fin_Tq
Ecrire (I) *{Résultat du classement entrelacé }*
Fin.

Figure V.3. Algorithme « Balanced interleaving » (Chapelle et al., 2012).

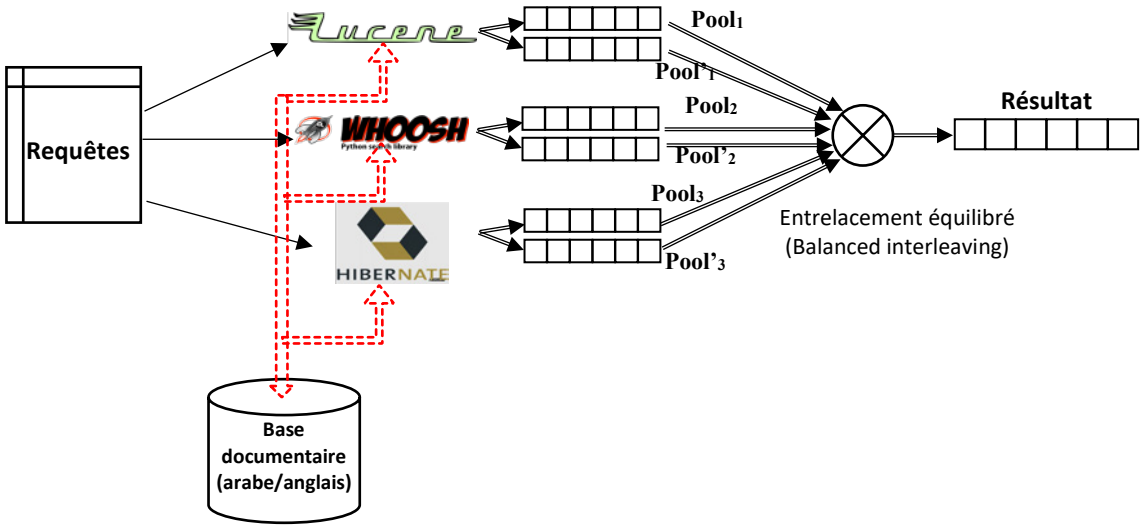

Figure V.4. Aperçu général sur la stratégie de Pooling.

Concernant le calcul des scores de ces pertinences, il se fait donc par le principe présenté dans la section suivante.

V.4.2. Calcul des scores des pertinences

Afin de calculer les scores des pertinences de chaque document par rapport aux requêtes, nous procédons comme suit ; au début, pour chaque requête donnée, nous sélectionnons les K au maximum premiers documents retournés par chaque moteur de recherche (la valeur de k=50 est déterminée expérimentalement), en fait, nous les considérons tous comme des documents pertinents avec un score de pertinence graduel. Le degré du score de pertinence est calculé selon l'appartenance du document au classement de la liste retournée. Le tableau V.7 détermine ce degré de pertinence par rapport aux intervalles du classement du document.

Tableau V.7. Score de pertinence selon le classement du document.

intervalle de classement du documents dans la liste retournée	**0%- 20%**	**20%- 40%**	**40%-60%**	**60%- 80%**	**80%- 100%**
Score de pertinence	1,0	0,8	0,6	0,4	0,2

Ensuite, la pertinence récupérée doit être validée avec deux conditions mutuelles. La première condition est la validation manuelle par l'expert humain "*que le document est vraiment en relation avec la requête"* (l'expertise humaine est effectuée par au moins deux étudiants masters/doctorants). Dans la deuxième condition, pour qu'un document soit accepté comme pertinent, il faut qu'il figure au moins dans quatre résultats de pools parmi les six retournés par les moteurs de recherches.

En conséquence, le score du degré de la pertinence du document est égal à la moyenne de six résultats des Pools. Le tableau V.8 explique le calcul final du score pour un échantillon de requêtes, où :

1. **Pool$_1$** : liste des documents retournés par **Lucene** pour la requête exprimée en **arabe**.
2. **Pool'$_1$**: liste des documents retournés par **Lucene** pour la requête exprimée en **anglais**.
3. **Pool$_2$** : liste des documents retournés par **Whoosh** pour la requête exprimée en **arabe**.
4. **Pool'$_2$**: liste des documents retournés par **Whoosh** pour la requête exprimée en **anglais**.
5. **Pool$_3$** : liste des documents retournés par **Hibernate** pour la requête exprimée en **arabe**.
6. **Pool'$_3$**: liste des documents retournés par **Hibernate** pour la requête exprimée en **anglais**.

Tableau V.8. Calcul du score de la pertinence.

Requête	Document	$Pool_1$	$Pool'_1$	$Pool_2$	$Pool'_2$	$Pool_3$	$Pool'_3$	Validation humaine	Score de pertinence (=moyenne)
Q001	D516	0.4	0.2	0.4	0.0	0.4	0.2	Oui	0.267
Q001	D103	0.2	0.2	0.0	0.6	0.0	0.4	Non	0.000
Q001	D091	0.2	0.6	0.4	0.8	0.2	0.6	Oui	0.467
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
Q165	**D342**	0.8	0.6	0.8	0.6	0.6	0.4	Oui	0.633
Q165	**D203**	0.8	1.0	0.8	1.0	0.8	0.8	Oui	0.867
Q165	**D490**	1.0	0.8	1.0	1.0	1.0	1.0	Oui	0.967
Q165	**D333**	1.0	1.0	1.0	1.0	1.0	1.0	Oui	1.000

V.4.3. Résultat de la pertinence par la stratégie de Pooling

Le tableau V.9 résume et rassemble les résultats précédents de la liste des requêtes, la liste des topics/sous-topics (tableau V.3) et les scores de pertinence calculés par la stratégie de Pooling après la validation humaine (tableau V.8). Au total, cette méthode a généré 3651 jugements de pertinence entre les 165 requêtes et les 632 documents de la collection.

Tableau V.9. Pertinence des documents.

Requête	Document	Topic	Sous-Topic	Score de pertinence
Q001	**D516**	T2	ST24	0,267
Q001	**D091**	T2	ST24	0,467
Q001	**D111**	T2	ST24	0,600
.	.	.	.	.
.	.	.	.	.
Q165	**D342**	T1	ST12	0,633
Q165	**D203**	T3	ST33	0,867
Q165	**D490**	T3	ST33	0,967
Q165	**D333**	T4	ST42	1,000

V.5. Pertinence par l'apprentissage automatique

Dans cette section, nous examinons les jugements de pertinence obtenus dans la phase précédente par la technique d'apprentissage automatique, il s'agit d'appliquer l'algorithme Naïve-Bayes pour la classification des documents selon leurs pertinences.

L'idée de base est d'entraîner le classificateur (Naïve-Bayes) sur une base documentaire qui regroupe les deux catégories ; les documents pertinents et les documents non pertinents, afin de créer un modèle, par la suite, nous testons ce modèle sur le reste des documents pour prédire leurs pertinences. L'expérimentation permet alors de mesurer l'efficacité de trouver la pertinence des documents par rapport aux requêtes.

Nous avons opté pour le classificateur Naïve-Bayes par rapport aux autres algorithmes d'apprentissage automatique parce que, premièrement, le Naïve-Bayes se base fondamentalement sur la technique du sac de mots et le principe de la fréquence des mots, dont ces deux techniques sont au cœur du principe de l'indexation des bases documentaires de la plupart des systèmes de RI. Deuxièmement, le classificateur Naïve-Bayes est largement utilisé dans la littérature pour la classification des textes, par exemple le filtrage des documents, la détection des spams, l'analyse des sentiments, etc.

V.5.1. Classificateur Naïve-Bayes

Le Naïve-Bayes est un classificateur probabiliste, ce qui signifie que pour un document d, parmi toutes les classes $c \in C$, le classificateur renvoie la classe $\hat{c}$ qui a la probabilité postérieure maximale pour un document donné, comme c'est présenté par la formule (V.2).

$$\hat{c} = argmax_{c \in C} P(c/d) \tag{V.2}$$

Plus formellement, le classificateur repose sur le théorème de Bayes (Bayes, 1763) donné par la formule (V.3) suivante.

$$P(x/y) = \frac{P(y/x)P(x)}{P(y)} \tag{V.3}$$

Où $P(x/y)$ est la probabilité conditionnelle d'un événement x sachant qu'un autre événement y de probabilité non nulle s'est réalisé.

Après le remplacement de la formule (V.3) dans la formule (V.2) nous obtiendrons la formule (V.4).

$$\hat{c} = argmax_{c \in C} P(c/d) = argmax_{c \in C} \frac{P(d/c)P(c)}{P(d)} \tag{V.4}$$

L'équation (V.4) est facilement simplifiée par la suppression du dénominateur *P(d)*, puisque sa valeur est une constante qui ne se change pas pour toutes les classes, ce qui donne la formule (V.5)

$$\hat{c} = argmax_{c \in C} P(c/d) = argmax_{c \in C}(d/c\ )P(c) \quad \text{(V.5)}$$

Par ailleurs, cette formule est adaptée aux documents qui sont considérés comme un ensemble de mots (m_i) ce qui déduit la formule (V.6) appelée par les estimations de vraisemblance maximale (Maximum-Likelihood).

$$\hat{P}(c) = argmax_{c \in C}\ P(c) \prod_i P(m_i|c) \quad \text{(V.6)}$$

Où, la probabilité de classes *P(c)* « pertinente/non pertinente » est calculée par la formule (V.7) suivante :

$$P(c) = \frac{N_C}{N} \quad \text{(V.7)}$$

Où *c* indique la classe (pertinente/non pertinente). *Nc* est égal au nombre de documents dans la classe (pertinente/non pertinente). *N* est égal au nombre total des documents dans la partie d'entraînement.

$P(m_i/c)$ est la probabilité du mot m_i dans le document sachant la classe *c* (pertinente/non pertinente), elle est calculée par la formule (V.8).

$$P(w_i|c) = \frac{Nombre(m_i,c)+1}{Nombre(c)+|v|} \quad \text{(V.8)}$$

Où : $Nombre(m_i,c)$ indique le nombre de mots m_i dans la classe *c* (pertinente/non pertinente). *Nombre(c)* est égal au nombre total des mots dans la classe *c* (pertinente/non pertinente). $|v|$ est égal au nombre total du vocabulaire de tous les documents dans la base d'entraînement.

V.5.2. Apprentissage automatique par le classificateur Naïve-Bayes

En pratique, le classificateur Naïve-Bayes est souvent utilisé dans la classification des documents. Concernant notre projet, nous créons une partie d'entraînement pour chaque requête, puis nous testons la pertinence sur les documents restants de la collection par l'application de l'algorithme multinomial de Naïve-Bayes, il consiste précisément à utiliser les estimations de vraisemblance maximale (Maximum-Likelihood) comme présentée dans la formule (V.6). En effet, pour classer un document, l'algorithme sélectionne entre deux

classes, la classe pertinente ou la classe non pertinente, il s'agit de la classe qui génère plus de probabilité calculée aussi par la formule (V.6).

Plus précisément, l'algorithme effectue une classification binaire des textes (pertinent/non pertinent), dans laquelle, un document est analysé comme un groupe de mots, où chaque mot est supposé être généré indépendamment aux autres mots (hypothèse du sac de mots). La Figure V.5. illustre cet algorithme de classification tel qu'il est décrit par (Jurafsky & Martin, 2019).

Algorithme 02 : Naïve-bayes

Fonction *Entrainement_NB (D,C)* **retourne** *(log(P(c)) , log(P(w/c))*
  Pour *chaque classe c ∈ C* **Faire** *{Calcul P(c) termes }*
    Début
      N_{doc} = *Nombre de documents dans D*
      N_c = *Nombre de documents de D dans la classe c*
      Logprior[c]$\leftarrow log \frac{N_c}{N_{doc}}$
      V ← *vocabulaire de D*
      bigdoc[c] ← **ajout**(d) **Pour** *d ∈ D* **avec** *classe c*
      Pour c*haque mot w ∈ V* **Faire** *{Calcul P(w/c) termes }*
        Début
          count(w,c) ← *Nombre d'occurrences de w dans bigdoc[c]*
          loglikelihood[w,c] ← $log \frac{count(w,c)+1}{\sum_{w' \in V} (count(w',c)+1)}$
        Fin_Pour
    Fin_Pour
  retourne *logprior, loglikelihood, V*

Fonction *Test_NB (testdoc, logprior, loglikelihood, C, V)* **retourne** *(meilleur c)*
Pour *chaque classe c ∈ C* **Faire**
    Début
      sum[c]← *logprior[c]*
      Pour *chaque position i dans testdoc* **Faire**
        Début
          mot ← *testdoc[i]*
          Si *mot ∈ V* **alors**
              sum[c]← *sum[c] + loglikelihood[mot,c]*
        Fin_si
        Fin_Pour
    Fin_Pour
retourne $argmax_c$ *sum[c]*

Figure V.5. Algorithme de classification Naïve-bayes (Jurafsky & Martin, 2019).

V.5.3. Mesures de performance

Afin d'évaluer les performances du classificateur Bayésien, nous utilisons les métriques relatives à l'apprentissage automatique, il s'agit de : la Précision (P), le Rappel (R) et la F-mesure (présentées dans le chapitre I).

- La précision est égale au nombre de documents réellement pertinents sélectionnés, divisé par le nombre de documents total sélectionnés par l'algorithme Bayésien comme pertinents.
- Le rappel est égal au nombre de documents réellement pertinents sélectionnés, divisé par le nombre total de documents pertinents dans la partie d'entraînement.
- La F-mesure est égale à la combinaison de la précision et du rappel déterminée par la formule (V.9).

$$F - mesure = \frac{2PR}{P+R} \quad \text{(V.9)}$$

V.5.4. Expérimentation par le classificateur Bayésien

Dans cette sous-section, nous examinons l'efficacité des performances du classificateur Bayésien afin de prédire la pertinence des documents. Pour l'implémentation nous avons utilisé la variante « Naïve-Bayes multinomial »

Nous savons bien que l'apprentissage automatique, est le processus de construction d'un modèle à partir des données exemples appelées la *partie d'entraînement* (Training, en anglais) et de tester les performances de ce modèle sur la *partie de test*, ces deux parties forment l'ensemble des données (Dataset), ou le corpus des documents dans notre cas d'étude.

Tout d'abord, nous commençons par la création de la partie d'entraînement en sélectionnant un tiers (1/3) des documents pertinents pour chaque requête, puis nous ajoutons aléatoirement un nombre égal des documents non pertinents depuis la collection, nous effectuons cette opération afin de concevoir une partie d'entraînement équilibrée (*Balanced Dataset*) entre les deux classes : pertinente et non pertinente. Puisqu'une partie d'entraînement équilibrée renforce le classificateur d'apprendre plus efficacement le modèle.

Ensuite, l'algorithme Naïve-Bayes est exécuté sur le reste du corpus afin de récupérer et sélectionner les documents pertinents, le classificateur prédit alors la classe des documents un par un en appliquant la formule (V.6). La classe est attribuée selon la probabilité la plus élevée (pertinent/non pertinent) comme c'est expliqué par l'algorithme présenté dans la Figure 2. Un échantillon de requêtes sur les résultats de la partie de test est présenté dans le tableau V.10.

Tableau V.10. Échantillon de requêtes sur les résultats de l'apprentissage automatique.

	1ère Technique : Stratégie de Pooling	2emeTechnique : Apprentissage automatique							
		Partie d'entraînement (Training)			Partie de test			Mesures	
Requête	Nombre de documents pertinents	(1/3) Nbre de docs pertinents	Nbre de docs non pertinents aléatoires	Ensemble d'entraînement (Training set)	(2/3) Nbre de docs pertinents	Nbre de docs sélectionnés	Nbre de docs pertinents sélectionnés	Rappel = Nbre Doc : Pert Selc / Pert	Précision = Nbre Doc : Pert Selct / Selct
Q001	44	15	15	30	29	25	18	62.07%	72.00%
Q002	42	14	14	28	28	26	19	67.86%	73.08%
Q003	34	11	11	22	23	29	13	56.52%	44.83%
Q004	38	13	13	26	25	36	17	68.00%	47.22%
Q005	29	10	10	20	19	26	09	47.37%	34.62%
Q006	40	13	13	26	27	12	11	40.74%	91.67%
Q007	23	08	08	16	15	16	08	53.33%	50.00%
Q008	14	05	05	10	09	11	05	55.56%	45.45%
Q009	34	11	11	22	23	17	13	56.52%	76.47%
Q010	26	09	09	18	17	15	11	64.71%	73.33%
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
Q163	27	09	09	18	18	11	09	50.00%	81.82%
Q164	39	13	13	26	26	13	11	42.31%	84.62%
Q165	26	09	09	18	17	11	09	52.94%	81.82%
Moyenne	22.13	7.38	7.38	14.76	**14.76**	**13.28**	**9.16**	**62.06%**	**68.98%**

Les valeurs moyennes pour le nombre des documents pertinents, des documents sélectionnés et des documents pertinents sélectionnés par rapport aux résultats de test des 165 requêtes par le classificateur Bayésien sont également illustrées dans la Figure V.6. suivante.

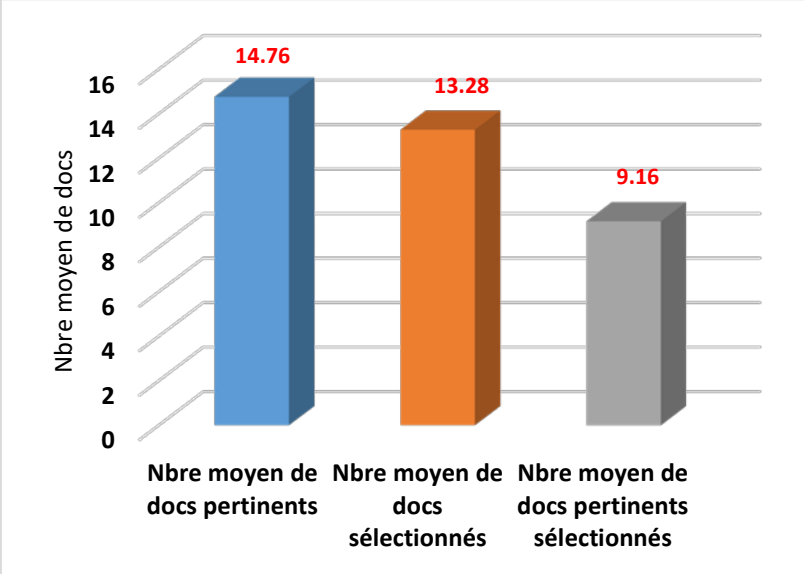

Figure V.6. Résultat de classification par Naïve-bayes.

Les mesures de performance de classification des documents pertinents par l'algorithme Bayésien sont indiquées dans le tableau V.11 ci-dessous. Nous rapportons cette performance par les mesures de précision, de rappel et de F-mesure.

Tableau V.11. Performance de l'apprentissage automatique.

Mesure	Valeur Moyenne
Précision	**68.98%**
Rappel	**62.06%**
F-mesure	**65.34%**

Nous rappelons que nous avons calculé la précision moyenne qui est égale à la moyenne des précisions de toutes les requêtes ou également au nombre moyen des documents pertinents sélectionnés divisé par le nombre moyen de documents sélectionnés (9,16/13,28= 0,6898 « **68.98%** »). Aussi, le Rappel moyen est égal à la moyenne des rappels

de toutes les requêtes ou également au nombre moyen de documents pertinents sélectionnés divisé par le nombre moyen de documents pertinents dans la partie de test (9,16/14,76 = 0,6206 « **62,06%** »). Tandis que, la F-mesure est calculée par la formule (V.5), et qui est égale à 0,6534 « **65,34%** ».

V.5.5. Expérimentation par word2vec et le classificateur Bayésien

A travers cette expérimentation, nous voulons tester la récupération de la pertinence par le classificateur Bayésien après avoir appliqué la technique du Word-Embbeding en utilisant le modèle du Word2vec sur la collection des documents.

L'objectif de cette technique est de récupérer les mots similaires par Word2vec et de les ajouter dans les textes, pour enrichir sémantiquement les documents.

V.5.5.1. Création du modèle word2vec

Le Word-Embedding (plongement du mot) est une représentation sémantique vectorielle des mots. Il est largement utilisé dans les applications du TAL récentes qui manipulent la sémantique. Le Word-Embedding est capable de capter le contexte d'un mot dans le texte, la relation avec les autres mots et la similarité sémantique et syntaxique.

Dans notre implémentation, nous commençons par la création des vecteurs Word-Embedding par l'algorithme Word2vec de l'apprentissage profond (Deep Learning), dans lequel le modèle Word2vec fournit deux types de modèles; le modèle CBOW (Continuous-Bag-Of-Words, en français Sac-de-mots-continu) qui consiste à prédire le mot cible à partir des mots de la fenêtre du contexte (Figure V.7), et le modèle Skip-Gram qui est l'opposé de CBOW, le Skip-gram est utilisé pour prédire les mots contextuels par rapport à un mot cible (Figure V.7). Dans cette expérimentation nous avons créé ces deux modèles par l'entraînement de l'algorithme word2vec sur un corpus composé de plus de 135.000 textes en arabe collectés aussi depuis les rapports des thèses.

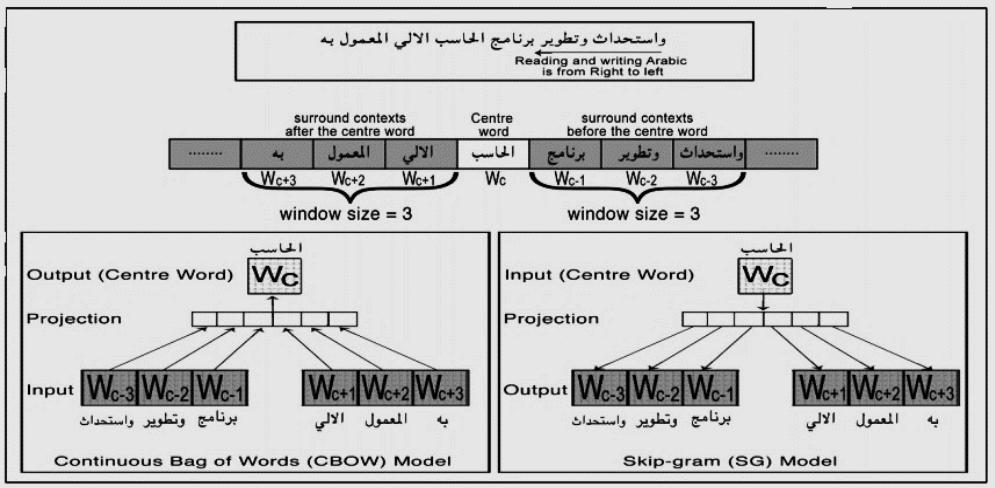

Figure V.7. Modèles CBOW et Skip-Gram du Word2vec (Alayba, Palade, England, & Iqbal, 2018).

Nous avons utilisé la bibliothèque Gensim[8] sous Python pour son implémentation, d'où, le Word2vec reçoit également des paramètres qui affectent la vitesse et la qualité de l'entraînement :

```
w2v=word2vec(vocab_proc, size=150, window=6, min_count=1, negative=10, iter=50, sg=1)
```

- **Dimension (size)** : signifie la taille du vecteur des mots ou le nombre de tokens utilisés pour représenter chaque mot. Dans notre cas nous avons utilisé la taille de 150 mots.
- **Taille de la fenêtre (window)** : c'est la distance maximale entre le mot cible et ses mots voisins à gauche ou à droite. Dans notre expérience nous avons testé pour les valeurs de N=3, N=6 et N=9.
- **Nombre d'itérations (iter) :** c'est le nombre de fois qu'un texte est passé par l'algorithme, plus en détail il signifie le nombre de passe avant et de passe arrière dans le réseau de neurones du modèle de l'apprentissage profond. Pour cette expérimentation le programme exécute 50 itérations.
- **Échantillonnage négatif (negative) :** c'est un paramètre pour mettre à jour le désordre dans les vecteurs de sortie recommandé égale à 10 par Mikolov (Mikolov, Chen, Corrado, & Dean, 2013).
- **sg** : 'sg = 1' pour le modèle Skip-gram et 'sg = 0' pour le modèle CBOW.

La Figure V.8 suivante montre un exemple du vecteur du mot (énergie - طاقة) créé par le modèle Skip-gram.

```
In [80]: print(w2v.wv["طاقة"])
[ 0.7040401  -1.2848525  -0.42216215  0.22770166  0.8033355   2.0475771  -0.2295513
 -0.2295513   1.8982042   0.82281342  1.5872442   0.39776078 -1.5901227   0.83469486
  0.83469486  0.7866394   0.19596177  1.0551784   0.9240165  -0.7113917  -0.55556214
 -0.55556214 -1.0121439   0.62308985 -0.2988863   0.26678622  0.13348942  0.66353184
  0.66353184 -1.7477108   1.51155233  0.17262405  0.79775923  0.04146605 -0.39128792
 -0.39128792  0.8509117   1.528466    1.2750385   0.44276437  1.1806506   0.4684516
  0.4684516   1.7765303  -0.30092058 -1.6178737   0.5582926  -1.6431113   1.3137595
  1.3137595   1.118497    0.00734752  0.95286727  0.5793374  -0.70991546 -1.4745585
 -1.4745585  -0.4003153   0.7183221  -2.6563544   1.4938401   2.6417506  -0.13973233
 -0.13973233  0.6432049  -1.5447314   0.5740125  -0.40174508  1.7563851  -2.5445292
 -2.5445292   1.0314955  -0.34289467 -0.15681833 -0.6840776  -1.0306265   1.7645836
  1.7645836  -1.0554788  -0.6250649  -0.73714274  2.190594    0.5345262   1.053404
  1.053404   -1.4098868   1.1140178  -0.4519419   1.3608686  -1.2253515   1.538713
```

Figure V.8. Exemple du vecteur du mot *« énergie – طاقة »*.

[8] https://pypi.org/project/gensim/

V.5.5.2. Enrichissement des documents par les mots similaires

Après la création des deux modèles, nous avons enrichi les documents de la collection par l'ajout des mots similaires directement dans les textes des documents. La similarité des mots est calculée par la formule de Cosinus (formule V.10) sur les vecteurs des mots créés par les deux modèles, le Skip-gram et le CBOW, où le seuil de similarité est supérieur ou égal à 60% (ce seuil est désigné expérimentalement).

$$\text{Similarité}(A, B) = \text{Cos}(A, B) = \frac{A.B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}} \qquad \text{(V.10)}$$

Le tableau V.12 suivant présente les mots similaires ainsi leurs valeurs de similarité calculées par la formule (V.10) pour le mot exemple (énergie - طاقة).

Tableau V.12. Mots similaires pour le mot exemple (énergie - طاقة).

N°	Mots similaires		similarité	N°	Mots similaires		similarité
	en arabe	en français			en arabe	en français	
1	البترول	Le pétrole	0,83053	21	نقاط	Points	0,39675
2	إستهلاك	Consommation	0,78165	22	نقل	Transfert	0,39564
3	جراء	à cause de	0,73437	23	طبقة	couche	0,35434
4	في	dans	0,72885	24	متعددة	Plusieurs	0,34708
5	الكهربائية	Electrique	0,72706	25	تحسين	amélioration	0,34651
6	خلال	Pendant	0,70355	26	خطوط	Lignes	0,34481
7	الضوئية	L'optique	0,70017	27	شبكات	Réseaux	0,33436
8	وانتقال	et transmission	0,69319	28	حيث	dont	0,32368
9	تعمل	Travaille	0,66517	29	ترانزستورات	Transistors	0,30306
10	موجات	Ondes/Vagues	0,59587	30	رقيقة	Puce/Mince	0,29196
11	تحكم	contrôle	0,57545	31	الاقتصادي	Economique	0,29064
12	الضوء	La lumière	0,56397	32	تصميم	Conception	0,25964
13	أداء	performance	0,54585				
14	المركبة	La composée	0,53168				
15	إستخدام	Utilisation	0,50861				
16	إمتصاص	Absorption	0,49555				
17	أنظمة	Systèmes	0,48372				
18	تحديد	Spécifier	0,40124				
19	خوارزميات	Algorithmes	0,39953				
20	سيليكون	Silicone	0,39802				

V.5.5.3. Résultat du test

Dans cette partie, nous examinons les résultats de récupération de la pertinence par le classificateur Naïve-Bayes appliqué sur la base documentaire enrichie par les deux modèles de Word2vec (le Skip-gram et le CBOW), dont l'algorithme Bayésien est exécuté sous les mêmes conditions présentées dans la section (V.5.4. Expérimentation par le classificateur Bayésien). Le tableau V.13 résume les résultats obtenus.

Tableau V.13. Résultat de performance du Naïve-Bayes avec le Word2vec.

Mesure	Base documentaire originale	Base documentaire enrichie par le Word2vec					
		Skip-gram			CBOW		
		N=3	N=6	N=9	N=3	N=6	N=9
Précision	68,98%	71,01%	71,40%	**72,48%**	69,17%	69,61%	69,61%
Rappel	62,06%	63,97%	64,59%	**65,50%**	62,26%	62,76%	62,71%
F-mesure	65,34%	67,30%	67,83%	**68,81%**	65,53%	66,01%	65,98%

D'après les résultats présentés dans le tableau V.13 et le graphe de la Figure V.9, nous déduisons ce qui suit :

- L'enrichissement de la base documentaire par les modèles du word2vec a renforcé les performances du classificateur Bayésien pour la récupération des documents pertinents.
- Les meilleurs résultats sont obtenus après l'enrichissement par le modèle Skip-gram par rapport à l'enrichissement par le modèle CBOW.
- Le paramètre de la fenêtre N=9 pour le modèle Skip-gram a permis au classificateur de réaliser la meilleure performance avec une F-mesure= 68.81%, donc une progression de 5,32% par rapport à la F-mesure de la base documentaire originale (65,34%).
- La taille de la fenêtre ou le nombre N de mots dans le contexte pour le Skip-Gram ou le CBOW ne différencie par beaucoup les performances du classificateur, puisque ses résultats sont variés entre 0,29% et 5,31%.

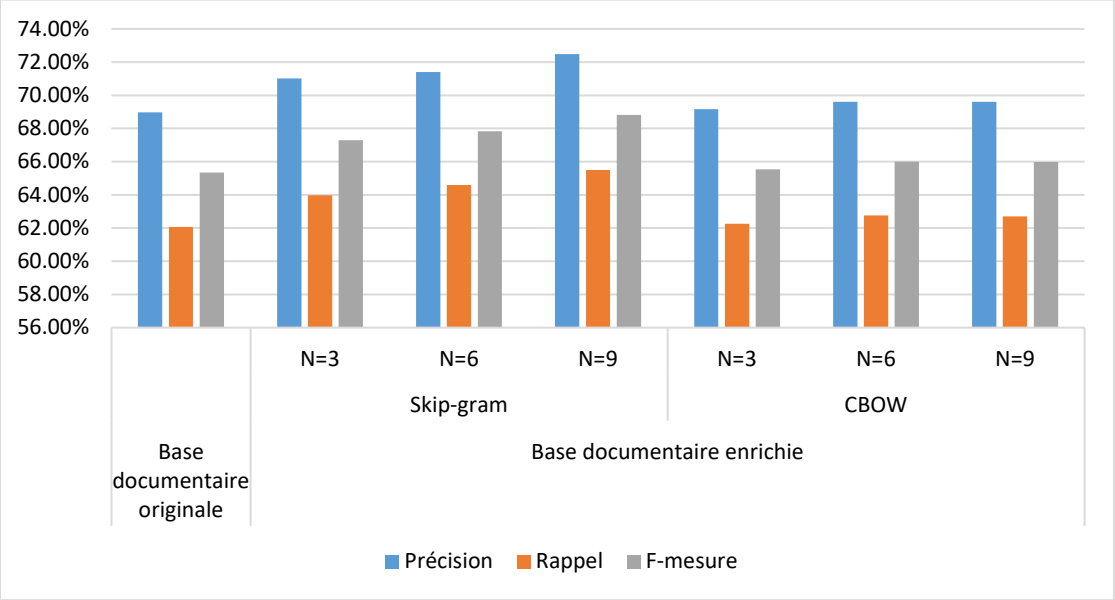

Figure V.9. Illustration graphique de performance du Naïve-Bayes avec le Word2vec.

V.6. Discussion et perspectives

Au début du travail, nous avons consacré de gros efforts à la collecte et à l'analyse des données, à la création de la liste des requêtes (165 requêtes), à la collecte des jugements de pertinence des requêtes par rapport aux documents par les experts (632x165 = 104.280 jugements de pertinence). Nous avons présenté ensuite les résultats expérimentaux pour montrer l'adéquation de l'utilisation des modèles d'apprentissage automatique pour récupérer ces jugements de pertinence. En analysant ces résultats, nous déduisons ce qui suit :

- D'après la valeur du rappel, qui est égale à près de 62% (0,6206), cela représente le nombre de documents qui sont effectivement sélectionnés comme des documents pertinents par le classificateur Naïve-Bayes par rapport à ceux qui sont choisis par la stratégie de Pooling et puis validés par des experts humains. Ce qui indique que la méthode de l'apprentissage automatique a pu sélectionner correctement deux tiers des documents pertinents d'une manière automatique.
- L'expérimentation a également montré des résultats probants, de 65,34% de F-mesure, suite à l'utilisation du classificateur de Naïve-Bayes, cela indique que ces techniques de l'apprentissage automatique peuvent classer et détecter automatiquement la pertinence des documents de manière efficace et efficiente.
- Le principal avantage de l'algorithme Naïve-Bayes est qu'il nécessite un petit nombre d'instances dans la partie d'apprentissage (Entraînement) pour estimer les paramètres nécessaires à la classification.

- De plus, par rapport aux travaux de l'état de l'art, cette étude soutient le travail de (Bounhas & Guirat, 2019) où la technique de classification par l'algorithme de l'apprentissage automatique du SVM (Support Vector Machine, en français : machine à vecteurs de support) a été utilisée, par laquelle, les auteurs ont créé un modèle équilibré composé de 17 requêtes pour la partie d'entraînement et la partie de test. Le SVM avait classé, selon les jugements de pertinence, des vecteurs de caractéristiques créés à partir les scores des documents sélectionnés par chaque système de RI. Les résultats expérimentaux ont montré pareillement une amélioration de la valeur de la précision moyenne MAP de 25,87% (bien entendu que la valeur de MAP ne généralise pas cette conclusion puisque le nombre des requêtes est réduit (17 requêtes) mais elle mentionne un signe positif sur la performance).
- En conséquence, les méthodes de l'apprentissage automatique aident certainement à créer plus efficacement et à moindre coût les collections de test de RI.
- L'enrichissement de la base documentaire par les modèles de word2vec à aider d'améliorer les performances du classificateur Bayésien de 5.32%, cela indique que la technique de Word-Embedding (plongement de mots) renforce les performances du classificateur.
- En suggestion, pour améliorer les scores des performances du classificateur, nous recommandons les propositions suivantes :
    - Augmentation de la taille du corpus pour la partie d'apprentissage et la partie de test. La disponibilité de plus de données permet d'obtenir des modèles meilleurs et précis.
    - Utilisation plus de techniques de pondération des mots comme Tf-Idf ou l'une de ses variantes pour pondérer les mots de la collection. Puisque des formules différentes peuvent produire des résultats différents.
    - Caractéristiques de la base documentaire: Pour le problème de classification, il est important de bien choisir les données de la partie d'entraînement et la partie de test. La variation des caractéristiques permet à l'algorithme de classification d'apprendre plus et améliore mieux ses performances.
    - Utilisation d'autres représentations sémantiques vectorielles récentes telles que Doc2vec, Sec2vec, Elmo, ou même par Transfer Learning à savoir BERT (à travers ses variantes ; AraBert, RoBERTA ou XLM-RoBERTa) ou GPT2.
- En perspective, les algorithmes de l'apprentissage automatique traditionnel (Machine Learning) et les algorithmes d'apprentissage profond (Deep Learning) sont plus susceptibles d'être la meilleure solution à l'avenir pour construire les grandes collections de RI.

- Comme travail futur, nous souhaitons rassembler plus de chercheurs autour de notre approche afin de créer une grande collection de RI pour la langue arabe, cette collection pourra devenir une importante ressource gratuite pour la communauté des chercheurs qui ressemble à celles de TREC pour la langue anglaise.

V.7. Conclusion

Nous avons présenté dans ce chapitre notre méthode de création de collections de test de recherche d'information. Par laquelle, nous avons montré que l'algorithme d'apprentissage automatique combiné à la stratégie de Pooling offre une solution efficace à moindre coût pour créer des jugements de pertinence.

Pour application, nous avons créé une collection composée de 632 documents bilingues parallèles (arabe/anglais) et de 165 requêtes. Les textes des documents sont collectés principalement depuis les résumés des thèses de doctorats et magisters hébergées dans les sites web des universités et aussi fournis par la bibliothèque en ligne *ProQuest*.

Nous avons mis notre collection de test de RI en ligne [9] pour libre accès et téléchargement gratuit, par laquelle elle fournit aux chercheurs une ressource pour évaluer et tester leurs systèmes de RI ou leurs nouvelles approches proposées.

En conclusion, nous avons démontré le potentiel de la méthode proposée par, tout d'abord, l'algorithme d'apprentissage automatique de Naïve-Bayes qui pourrait sélectionner les documents pertinents avec une performance de F-mesure égale à 65.34%. Deuxièmement, l'enrichissement sémantique des documents par les modèles word2vec a aidé l'amélioration les performances du classificateur Bayésien. Troisièmement, un exemple concret de la collection de test de RI a été créé.

Les résultats suggèrent que la méthode proposée pourrait également être très utile pour aider à créer des nouvelles grandes collections de test de recherche d'information à moindre coût par rapport aux autres méthodes précédentes.

[9] https://sourceforge.net/projects/arabic-english-ir-collection/

Conclusion générale

Ils existent peu de travaux qui se sont intéressés au domaine du traitement automatique de la langue arabe et plus précisément à la recherche d'information, pourtant elle fait partie des cinq langues les plus parlées au monde, avec plus de 300 millions de locuteurs natifs selon les nations unies (2021).

A nos jours, de plus en plus des bases documentaires en arabe sont créées à travers les bibliothèques ou via le web, ce qui met les systèmes de recherche d'information et les moteurs de recherche en général en défit pour traiter la problématique de recherche d'information en arabe.

A travers cette thèse, nous avons apporté des solutions à cette problématique par deux contributions.

Premièrement, nous avons proposé une nouvelle approche pour la recherche d'information sémantique arabe par la reformulation des requêtes grâce à des arbres sémantiques. Le principe de l'approche est de construire un arbre sémantique à partir des mots-clés originaux de la requête initiale, le processus de génération de cet arbre passe par l'initialisation des concepts de bases et puis ajoute les autres concepts à travers des extensions par des relations sémantiques, telle que la synonymie, l'hyponymie et l'hyperonymie. La méthode proposée utilise principalement la ressource du WordNet arabe pour la désambiguïsation des concepts et la hiérarchisation de l'arbre. Une fois l'arbre sémantique est généré, le processus reformule enfin la requête initiale par l'ajout des nouveaux mots-clés pondérés extraits de cet arbre. L'approche a été implémentée et les résultats de test ont montré une amélioration dans les performances autour de 10% de précision moyenne MAP (*Mean Average Precision*).

Deuxièmement, nous avons proposé une nouvelle méthode pour la création d'une collection de test de RI arabe, cette méthode combine la technique de la stratégie de Pooling utilisant les moteurs de recherches et l'algorithme de Naïve-Bayes de l'apprentissage automatique. Pour l'expérimentation, nous avons créé une nouvelle collection de test de RI. Cette collection est composée d'une liste de 165 requêtes avec leurs jugements de pertinence et un corpus textuel parallèle formé de 632 documents arabes et 632 documents anglais.

Parmi les apports de notre thèse est la mise en ligne[1] de cette collection pour libre accès, par laquelle elle offre aux utilisateurs une ressource intéressante, afin de tester et d'évaluer leurs systèmes de RI ou d'autres algorithmes tels que les algorithmes de classification ou de catégorisation des documents.

A travers cette deuxième contribution, nous avons également confirmé l'adéquation de l'utilisation des méthodes de l'apprentissage automatique à savoir le Naïve-Bayes pour pouvoir repérer les documents pertinents relatifs aux requêtes avec une performance de F-mesure égale à 65.34%. En conclusion, ces méthodes de l'apprentissage automatique servent sans aucun doute à créer des grandes collections de RI de manière plus efficace par rapport aux méthodes coûteuses précédentes.

En plus, nous avons aussi montré que l'intégration des techniques du Word-Embedding (plongement de mots) pour l'enrichissement sémantique des textes, telles que le word2vec, renforce efficacement les résultats de l'algorithme de l'apprentissage automatique pour récupérer les jugements de pertinence des documents.

Dans le futur, nous souhaitons réunir plus de chercheurs autour de notre proposition afin de créer une grande collection de test de RI pour la langue arabe, cette dernière pourra devenir une importante ressource gratuite pour la communauté des chercheurs qui ressemble à celles du TREC pour la langue anglaise.

En perspective, nous voulons aussi migrer plus vers les algorithmes de Deep Learning (exemple : RNN, LSTM, GRU, Bi-GRU…), Transfer Learning (exemple : GPT2, GPT3 ou BERT par ses variantes comme AraBERT, RoBERTa ou XLM-RoBERTa) et les techniques du Word-Embedding par les représentations sémantiques vectorielles récentes telles que Doc2vec, Sec2vec, sentence2vec,…etc. afin de construire des éventuelles grandes collections de test de RI arabe, par exemple par la génération automatique des bases documentaires en utilisant le principe d'augmentation de données « *Data Augmentation ».*

[1] https://sourceforge.net/projects/arabic-english-ir-collection/

Bibliographie

Abbache, A., Meziane, F., Belalem, G., & Belkredim, F. Z. (2018). Arabic query expansion using wordnet and association rules. In *Information Retrieval and Management: Concepts, Methodologies, Tools, and Applications* (pp. 1239–1254). IGI Global.

Abderrahim, M. E. A. (2014). Concept based vs. pseudo relevance feedback performance evaluation for information retrieval system. *ArXiv Preprint ArXiv:1403.4362*.

Abu-Salem, H., Al-Omari, M., & Evens, M. W. (1999). Stemming methodologies over individual query words for an Arabic information retrieval system. *Journal of the American Society for Information Science*, *50*(6), 524–529.

Aggarwal, N., & Buitelaar, P. (2012). Query expansion using wikipedia and DBpedia. *CLEF (Online Working Notes/Labs/Workshop)*.

Agirre, E., & Edmonds, P. (2007). *Word sense disambiguation: Algorithms and applications* (Vol. 33). Springer Science & Business Media.

Aklouche, B., Bounhas, I., & Slimani, Y. (2018). Query Expansion Based on NLP and Word Embeddings. *TREC*.

Aklouche, B., Bounhas, I., & Slimani, Y. (2019). Pseudo-Relevance Feedback Based on Locally-Built Co-occurrence Graphs. In *European Conference on Advances in Databases and Information Systems* (pp. 105–119). https://doi.org/10.1007/978-3-030-28730-6_7

Al-Kharashi, I. A., & Evens, M. W. (1994). Comparing words, stems, and roots as index terms in an Arabic information retrieval system. *Journal of the American Society for Information Science*, *45*(8), 548–560.

Alayba, A. M., Palade, V., England, M., & Iqbal, R. (2018). Improving Sentiment Analysis in Arabic Using Word Representation. *2nd IEEE International Workshop on Arabic and Derived Script Analysis and Recognition, ASAR 2018*, 13–18. https://doi.org/10.1109/ASAR.2018.8480191

Aljlayl, M., Beitzel, S. M., Jensen, E. C., Chowdhury, A., Holmes, D. O., Lee, M., … Frieder, O. (2001). IIT at TREC-10. *TREC*.

Alnaied, A., Elbendak, M., & Bulbul, A. (2020). An intelligent use of stemmer and morphology analysis for Arabic information retrieval. *Egyptian Informatics Journal*, *21*(4), 209–217.

Althobaiti, M., Kruschwitz, U., & Poesio, M. (2014). *AraNLP: A Java-based library for the processing of Arabic text*.

Annamoradnejad, I., & Zoghi, G. (2020). Colbert: Using bert sentence embedding for humor detection. *ArXiv Preprint ArXiv:2004.12765*.

Antoun, W., Baly, F., & Hajj, H. (2020). Arabert: Transformer-based model for arabic language understanding. *ArXiv Preprint ArXiv:2003.00104*.

Aseervatham, S. (2009). A concept vector space model for semantic kernels. *International Journal on Artificial Intelligence Tools*, *18*(02), 239–272.

Attia, Mohamed, Rashwan, M. A. A., & Al-Badrashiny, M. A. (2009). Fassieh, a semi-automatic visual interactive tool for morphological, PoS-Tags, phonetic, and semantic annotation of Arabic Text Corpora. *IEEE Transactions on Audio, Speech, and Language Processing*, *17*(5), 916–925.

Attia, Mohammed, Toral, A., Tounsi, L., Monachini, M., & van Genabith, J. (2010). *An automatically built named entity lexicon for Arabic*.

Banerjee, S., & Pedersen, T. (2002). An adapted Lesk algorithm for word sense disambiguation using WordNet. *International Conference on Intelligent Text Processing and Computational Linguistics*, 136–145. Springer.

Bartunov, S., Kondrashkin, D., Osokin, A., & Vetrov, D. (2016). Breaking sticks and ambiguities with adaptive skip-gram. *Artificial Intelligence and Statistics*, 130–138. PMLR.

Bayes, T. (1763). An Essay Toward Solving a Problem in the Doctrine of Chances. *Reprinted in Facsimiles of Two Papers by Bayes. Hafner. Publishing (1963)*, *53*.

Baziz, M., Boughanem, M., & Aussenac-Gilles, N. (2005). Conceptual indexing based on document content representation. *International Conference on Conceptions of Library and Information Sciences*, 171–186. Springer.

Baziz, M., Boughanem, M., & Traboulsi, S. (2005). A concept-based approach for indexing documents in IR. *INFORSID*, *2005*, 489–504. Citeseer.

Benajiba, Y., Rosso, P., & Benedíruiz, J. M. (2007). Anersys: An arabic named entity recognition system based on maximum entropy. *International Conference on Intelligent Text Processing and Computational Linguistics*, 143–153. Springer.

Bengio, Y., Ducharme, R., Vincent, P., & Janvin, C. (2003). A neural probabilistic language model. *The Journal of Machine Learning Research*, *3*, 1137–1155.

Black, W., Elkateb, S., Rodriguez, H., Alkhalifa, M., Vossen, P., Pease, A., & Fellbaum, C. (2006). Introducing the Arabic wordnet project. *Proceedings of the Third International WordNet Conference*, 295–300. Citeseer.

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, *5*, 135–146.

Boubekeur, F., & Azzoug, W. (2013). Pondération des Concepts en Indexation Sémantique. *CORIA'13: Dixième Édition de La Conférence En Recherche d'Information et Applications*. https://doi.org/10.24348/coria.2013.coria2013_44

Boubekeur, F., Boughanem, M., & Tamine-Lechani, L. (2007). Semantic information retrieval based on CP-nets. *2007 IEEE International Fuzzy Systems Conference*, 1–7. IEEE.

Bounhas, I., & Guirat, S. Ben. (2019). KUNUZ: A Multi-Purpose Reusable Test Collection for Classical Arabic Document Engineering. *2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA)*, 1–8. Abu Dhabi, United Arab Emirates: IEEE.

Bounhas, I., & Guirat, S. Ben. (2019). KUNUZ: A Multi-Purpose Reusable Test Collection for Classical Arabic Document Engineering. *2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA)*, 1–8. Abu Dhabi, United Arab Emirates: IEEE.

Bounhas, I., Soudani, N., & Slimani, Y. (2020). Building a morpho-semantic knowledge graph for Arabic information retrieval. *Information Processing & Management*, *57*(6), 102124. https://doi.org/10.1016/j.ipm.2019.102124

Bsoul, Q. W., & Mohd, M. (2011). Effect of ISRI stemming on similarity measure for Arabic document clustering. *Asia Information Retrieval Symposium*, 584–593. Springer.

Budanitsky, A., & Hirst, G. (2006). Evaluating wordnet-based measures of lexical semantic relatedness. *Computational Linguistics*, *32*(1), 13–47.

Cao, G., Nie, J.-Y., Gao, J., & Robertson, S. (2008). Selecting good expansion terms for pseudo-relevance feedback. *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 243–250.

Carpineto, C., & Romano, G. (2012). A survey of automatic query expansion in information retrieval. *ACM Computing Surveys (CSUR)*, *44*(1), 1–50. https://doi.org/10.1145/2071389.2071390

Carpineto, C., Romano, G., & Bordoni, F. U. (2004). Exploiting the potential of concept lattices for information retrieval with CREDO. *J. Univers. Comput. Sci.*, *10*(8), 985–1013.

Carterette, B., Pavlu, V., Kanoulas, E., Aslam, J. A., & Allan, J. (2008). Evaluation over thousands of queries. *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 651–658.

Chapelle, O., Joachims, T., Radlinski, F., & Yue, Y. (2012). Large-scale validation and analysis of interleaved search evaluation. *ACM Transactions on Information Systems*, *30*(1). https://doi.org/10.1145/2094072.2094078

Chen, A., & Gey, F. (2002). Building an Arabic stemmer for information retrieval. *TREC*, *2002*, 631–639.

Chen, Z., & Eickhoff, C. (2021). The Cross-Lingual Arabic Information REtrieval (CLAIRE) System. *ArXiv Preprint ArXiv:2107.13751*.

Clark, E. V, & MacWhinney, B. (1987). The principle of contrast: A constraint on language acquisition. *Mechanisms of Language Acquisition*, 1–33.

Colace, F., De Santo, M., Greco, L., & Napoletano, P. (2015). Weighted Word Pairs for query expansion. *Information Processing and Management*, *51*(1), 179–193. https://doi.org/10.1016/j.ipm.2014.07.004

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, *12*(ARTICLE), 2493–2537.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Stoyanov, V. (2019). Unsupervised cross-lingual representation learning at scale. *ArXiv Preprint ArXiv:1911.02116*.

Culpepper, J. S., Diaz, F., & Smucker, M. D. (2018). Research frontiers in information retrieval: Report from the third strategic workshop on information retrieval in lorne (swirl 2018). *ACM SIGIR Forum*, *52*(1), 34–90. ACM New York, NY, USA.

Da Costa, L. M., & Bond, F. (2016). Wow! What a useful extension! Introducing non-referential concepts to WordNet. *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, 4323–4328.

Darwish, K., & Ali, A. (2012). Arabic retrieval revisited: Morphological hole filling. *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 218–222.

Darwish, K., & Magdy, W. (2014). Arabic information retrieval. *Foundations and Trends® in Information Retrieval*, *7*(4), 239–342.

Darwish, K., & Oard, D. W. (2002). Term selection for searching printed Arabic. *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 261–268.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, *41*(6), 391–407.

Dilekh, T., Benharzallah, S., & Behloul, A. (2018). The Impact of Online Indexing in Improving Arabic Information Retrieval Systems. *Informatica*, *42*(4).

Dinh, B.-D. (2012). *Accès à l'information biomédicale: vers une approche d'indexation et de recherche d'information conceptuelle basée sur la fusion de ressources termino-ontologiques*. Université de Toulouse, Université Toulouse III-Paul Sabatier.

Dinh, D., & Tamine, L. (2010). Vers un modèle d'indexation sémantique adapté aux dossiers médicaux de patients. *Conférence Francophone En Recherche d'Information et Applications, CORIA 2010*, 325–336.

Egozi, O., Markovitch, S., & Gabrilovich, E. (2011). Concept-based information retrieval using explicit semantic analysis. *ACM Transactions on Information Systems*, *29*(2), 1–34. https://doi.org/10.1145/1961209.1961211

El Kah, A., & Zeroual, I. (2021). Arabic Topic Identification: A Decade Scoping Review. *E3S Web of Conferences*, *297*, 1058. EDP Sciences.

El Mahdaouy, A., El Alaoui, S. O., & Gaussier, E. (2018). Improving Arabic information retrieval using word embedding similarities. *International Journal of Speech Technology*, *21*(1), 121–136. https://doi.org/10.1007/s10772-018-9492-y

Elkateb, S., Black, W., Vossen, P., Rodríguez, H., Pease, A., Alkhalifa, M., & Fellbaum, C. (2006). Building a WordNet for Arabic. *Proceedings of the 5th International Conference on Language Resources and Evaluation, LREC 2006*, 29–34.

El-khair, I. A. (2006). Effects of Stop Words Elimination for Arabic Information Retrieval : A Comparative Study. *International Journal of Computing & Information Sciences*, *4*(3), 119–133.

Esuli, A., & Sebastiani, F. (2006). Sentiwordnet: A publicly available lexical resource for opinion mining. *LREC*, *6*, 417–422. Citeseer.

Farghaly, A., & Shaalan, K. (2009). Arabic natural language processing: Challenges and solutions. *ACM Transactions on Asian Language Information Processing (TALIP)*, *8*(4), 1–22.

Fellbaum, C. (1998). WordNet: An Electronic Lexical Database. *MIT Press, Cambridge*.

Gauch, S., Madrid, J. M., Induri, S., Ravindran, D., & Chadlavada, S. (2004). Keyconcept: A conceptual search engine. *Information and Telecommunication Technology Center, Technical Report: ITTC-FY2004-TR-8646*, *37*.

Gómez, J. M., Buscaldi, D., Rosso, P., & Sanchis Arnal, E. (2007). *JIRS Language-independent Passage Retrieval system: A comparative study*.

Guirat, S. Ben, Bounhas, I., & Slimani, Y. (2016). Combining indexing units for arabic information retrieval. *International Journal of Software Innovation (IJSI)*, *4*(4), 1–14.

Guirat, S. Ben, Bounhas, I., & Slimani, Y. (2019). Pre-indexing Techniques in Arabic Information Retrieval. *ICAART (2)*, 237–246.

Habash, N. Y. (2010). Introduction to Arabic natural language processing. In *Synthesis Lectures on Human Language Technologies* (Vol. 3). https://doi.org/10.2200/S00277ED1V01Y201008HLT010

Harrag, F., Hamdi-Cherif, A., Al-Salman, A. M. S., & El-Qawasmeh, E. (2009). Experiments in improvement of Arabic information retrieval. *3rd International Conference on Arabic Language Processing (CITALA), Rabat, Morocco*, 71–81.

Harrag, F., Hamdi-Cherif, A., Al-Salman, A. M. S., & El-Qawasmeh, E. (2011). Evaluating the effectiveness of VSM model and topic segmentation in retrieving arabic documents. *Computer Systems Science and Engineering*, *26*(1), 59–71.

Harrathi, F., Roussey, C., Maisonnasse, L., & Calabretto, S. (2010). Vers une approche statistique pour l'indexation sémantique des documents multilingues. *28ème Congrès INFORSID*, p-127.

Harris, Z. S. (1954). Distributional structure. *Word*, *10*(2–3), 146–162.

Hernandez, N., Hubert, G., Mothe, J., & Ralalason, B. (2008). *Recherche d'Information et Ontologies*. Universite de Toulouse Paul Sabatier III.

Hill, F., Reichart, R., & Korhonen, A. (2015). Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, *41*(4), 665–695.

Hmeidi, I., Kanaan, G., & Evens, M. (1997). Design and implementation of automatic indexing for information retrieval with Arabic documents. *Journal of the American Society for Information Science*, *48*(10), 867–881.

Hofmann, T. (1999). Probabilistic latent semantic indexing. *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 50–57.

Hope, D., & Keller, B. (2013). Maxmax: a graph-based soft clustering algorithm applied to word sense induction. *International Conference on Intelligent Text Processing and Computational Linguistics*, 368–381. Springer.

Howard, J., & Ruder, S. (2018). Universal language model fine-tuning for text classification. *ArXiv Preprint ArXiv:1801.06146*.

Hu, J., Deng, W., & Guo, J. (2006). Improving retrieval performance by global analysis. *18th International Conference on Pattern Recognition (ICPR'06)*, *2*, 703–706. IEEE.

Huang, G., Wang, S., & Zhang, X. (2011). Query expansion based on associated semantic space. *Journal of Computers*, *6*(2), 172–177. https://doi.org/10.4304/jcp.6.2.172-177

Ide, N., & Véronis, J. (1998). Word sense disambiguation: The state of the art. *Computational Linguistics*, *24*(1), 1–40.

Islam, A., & Inkpen, D. (2008). Semantic text similarity using corpus-based word similarity and string similarity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, *2*(2), 1–25.

Jones, K. S. (1971). *Automatic Keyword Classification for Information Retrieval*. London: Archon Books.

Jurafsky, D., & Martin, J. H. (2019). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition* (Book-3d-Ed). Stanford.

Kakadiaris, I., Paliouras, G., & Krithara, A. (2018). Proceedings of the 6th BioASQ Workshop A challenge on large-scale biomedical semantic indexing and question answering. *Proceedings of the 6th BioASQ Workshop A Challenge on Large-Scale Biomedical Semantic Indexing and Question Answering*.

Karisani, P., Rahgozar, M., & Oroumchian, F. (2016). A query term re-weighting approach using document similarity. *Information Processing & Management*, *52*(3), 478–489.

Khan, L., McLeod, D., & Hovy, E. (2004). Retrieval effectiveness of an ontology-based model for information selection. *The VLDB Journal*, *13*(1), 71–85.

Khatib, A. S. (1997). Terminological specifications and applications in the Arabic language. *Cultural Fifteenth Season of the Arabic Language Academy of Jordan*, 177–213. Amman, Jordan.

Khoja, S., & Garside, R. (1999). Stemming arabic text. *Lancaster, UK, Computing Department, Lancaster University*. https://doi.org/http://zeus.cs.pacificu.edu/shereen/research.htm

Khoury, R. (2011). Query classification using Wikipedia. *International Journal of Intelligent Information and Database Systems*, *5*(2), 143–163. https://doi.org/10.1504/IJIIDS.2011.038969

Klein, D., Toutanova, K., Ilhan, H. T., Kamvar, S. D., & Manning, C. D. (2002). Combining heterogeneous classifiers for word sense disambiguation. *Proceedings of the ACL-02 Workshop on Word Sense Disambiguation: Recent Successes and Future Directions*, 74–80.

Köhler, J., Philippi, S., Specht, M., & Rüegg, A. (2006). Ontology based text indexing and querying for the semantic web. *Knowledge-Based Systems*, *19*(8), 744–754.

Kutuzov, A., Dorgham, M., Oliynyk, O., Biemann, C., & Panchenko, A. (2018). Learning graph embeddings from WordNet-based similarity measures. *ArXiv Preprint ArXiv:1808.05611*.

Larkey, L. S., & Connell, M. E. (2001). Arabic information retrieval at UMass in TREC-10. *TREC*. Citeseer.

Larkey, L. S., Ballesteros, L., & Connell, M. E. (2002). Improving stemming for Arabic information retrieval: light stemming and co-occurrence analysis. *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 275–282.

Leacock, C., & Chodorow, M. (1998). Combining local context and WordNet similarity for word sense identification. *WordNet: An Electronic Lexical Database*, *49*(2), 265–283.

Lee, Y. K., & Ng, H. T. (2002). An empirical evaluation of knowledge sources and learning algorithms for word sense disambiguation. *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, 41–48.

Lesk, M. (1986). Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. *Proceedings of the 5th Annual International Conference on Systems Documentation*, 24–26.

Li, J., Chen, X., Hovy, E., & Jurafsky, D. (2015). Visualizing and understanding neural models in nlp. *ArXiv Preprint ArXiv:1506.01066*.

Li, W., & McCallum, A. (2006). Pachinko allocation: DAG-structured mixture models of topic correlations. *Proceedings of the 23rd International Conference on Machine Learning*, 577–584.

Li, Yinghao, Luk, W. P. R., Ho, K. S. E., & Chung, F. L. K. (2007). Improving weak ad-hoc queries using wikipedia asexternal corpus. *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'07*, 797–798. https://doi.org/10.1145/1277741.1277914

Li, Yuhua, McLean, D., Bandar, Z. A., O'shea, J. D., & Crockett, K. (2006). Sentence similarity based on semantic nets and corpus statistics. *IEEE Transactions on Knowledge and Data Engineering*, *18*(8), 1138–1150.

Lin, F., & Sandkuhl, K. (2008). A survey of exploiting wordnet in ontology matching. *IFIP International Conference on Artificial Intelligence in Theory and Practice*, 341–350. Springer.

Liu, L., Cao, C. ., Zhang, C. ., & Tian, G. . (2009). Sense recognition research of hyponymy based on concept space. *Chinese Journal of Computers*, *32*(8), 1651–1659.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., … Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *ArXiv Preprint ArXiv:1907.11692*.

Losada, D. E., Parapar, J., & Barreiro, A. (2018). Cost-effective construction of Information Retrieval test collections. *Proceedings of the 5th Spanish Conference on Information Retrieval*, 1–2. Zaragoza, Spain.

Maamouri, M., Bies, A., Buckwalter, T., & Mekki, W. (2004). The penn arabic treebank: Building a large-scale annotated arabic corpus. *NEMLAR Conference on Arabic Language Resources and Tools*, *27*, 466–467. Cairo.

Mahgoub, A., Rashwan, M., Raafat, H., Zahran, M., & Fayek, M. (2014). Semantic query expansion for Arabic information retrieval. *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, 87–92.

Maisonnasse, L., Gaussier, E., & Chevallet, J.-P. (2009). Model fusion in conceptual language modeling. *European Conference on Information Retrieval*, 240–251. Springer.

Mallak, I. (2011). *De nouveaux facteurs pour l'exploitation de la sémantique d'un texte en recherche d'information*. Université de Toulouse.

Mallat, S., Zouaghi, A., Hkiri, E., & Zrigui, M. (2013). Method of lexical enrichment in information retrieval system in Arabic. *International Journal of Information Retrieval Research (IJIRR)*, *3*(4), 35–51.

Mandreoli, F., Martoglia, R., & Tiberio, P. (2002). A syntactic approach for searching similarities within sentences. *Proceedings of the Eleventh International Conference on Information and Knowledge Management*, 635–637.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J. R., Bethard, S., & McClosky, D. (2014). The Stanford CoreNLP natural language processing toolkit. *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 55–60.

Manning, D. C., Prabhakar, R., & Hinrich, S. (2009). *An Introduction to Information Retrieval*. Cambridge university press.

Mayfield, J., McNamee, P., Costello, C., Piatko, C., & Banerjee, A. (2002). JHU/APL at TREC 2001: Experiments in filtering and in Arabic, video, and web retrieval. *AUTHOR Voorhees, Ellen M., Ed.; Harman, Donna K., Ed. TITLE The Text REtrieval Conference (TREC-2001)(10th, Gaithersburg, Maryland, November 13-16, 2001). NIST Special*, *500*, 59. Citeseer.

McCrae, J. P., & Prangnawarat, N. (2016). Identifying Poorly-Defined Concepts in WordNet with Graph Metrics. *International Semantic Web Conference*, 66–75. Springer.

McCrae, J. P., Wood, I., & Hicks, A. (2017). The Colloquial WordNet: Extending Princeton WordNet with Neologisms. *International Conference on Language, Data and Knowledge*, 194–202. Springer.

Melamud, O., Goldberger, J., & Dagan, I. (2016). context2vec: Learning generic context embedding with bidirectional lstm. *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, 51–61.

Mihalcea, R., Corley, C., & Strapparava, C. (2006). Corpus-based and knowledge-based measures of text semantic similarity. *Aaai*, *6*(2006), 775–780.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings*, 1–12.

Miller, G. A. (1995). WordNet: a lexical database for English. *Communications of the ACM*, *38*(11), 39–41.

Modha, S., & Majumder, P. (2019). An empirical evaluation of text representation schemes on multilingual social web to filter the textual aggression. *ArXiv Preprint ArXiv:1904.08770*.

Moro, A., Raganato, A., & Navigli, R. (2014). Entity linking meets word sense disambiguation: a unified approach. *Transactions of the Association for Computational Linguistics*, *2*, 231–244.

Navigli, R. (2009). Word sense disambiguation: A survey. *ACM Computing Surveys (CSUR)*, *41*(2), 1–69.

Navigli, R. (2016). Ontologies. In *The Oxford Handbook of Computational Linguistics 2nd edition*. https://doi.org/10.1093/oxfordhb/9780199573691.013.41

Navigli, R., & Ponzetto, S. P. (2012). BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, *193*, 217–250.

Navigli, R., Jurgens, D., & Vannella, D. (2013). Semeval-2013 task 12: Multilingual word sense disambiguation. *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, 222–231.

Neji, S., Jemni Ben Ayed, L., Chenaina, T., & M Shoeb, A. (2021). A Novel Conceptual Weighting Model for Semantic Information Retrieval. *Information Sciences Letters*, *10*(1), 14.

Nentidis, A., Krithara, A., Bougiatiotis, K., Krallinger, M., Rodriguez-Penagos, C., Villegas, M., & Paliouras, G. (2020). Overview of bioasq 2020: The eighth bioasq challenge on large-scale biomedical semantic indexing and question answering. *International Conference of the Cross-Language Evaluation Forum for European Languages*, 194–214. Springer.

Nirenburg, S., Domashnev, C., & Grannes, D. J. (1993). Two approaches to matching in example-based machine translation. *In Proc. of the 5th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI-93*. Citeseer.

Pal, D., Mitra, M., & Datta, K. (2014). Improving query expansion using WordNet. *Journal of the Association for Information Science and Technology*, *65*(12), 2469–2478. https://doi.org/10.1002/asi.23143

Pal, A. R., & Saha, D. (2015). Word sense disambiguation: A survey. *ArXiv Preprint ArXiv:1508.01346*.

Pantel, P., & Lin, D. (2002). Discovering word senses from text. *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 613–619.

Pasha, A., Al-Badrashiny, M., Diab, M. T., El Kholy, A., Eskander, R., Habash, N., … Roth, R. (2014). Madamira: A fast, comprehensive tool for morphological analysis and disambiguation of arabic. *LREC*, *14*(2014), 1094–1101. Citeseer.

Pathak, A. R., Agarwal, B., Pandey, M., & Rautaray, S. (2020). Application of deep learning approaches for sentiment analysis. In *Deep Learning-Based Approaches for Sentiment Analysis* (pp. 1–31). Springer.

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. *ArXiv Preprint ArXiv:1802.05365*.

Priss, U. (2000). Lattice-based information retrieval. *KO Knowledge Organization*, *27*(3), 132–142.

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). *Improving language understanding by generative pre-training*.

Radlinski, F., & Craswell, N. (2010). Comparing the sensitivity of information retrieval metrics. *Proceeding of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR '10*, 667. https://doi.org/10.1145/1835449.1835560

Reisinger, J., & Mooney, R. (2010). Multi-prototype vector-space models of word meaning. *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 109–117.

Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. *ArXiv Preprint Cmp-Lg/9511007*.

Rothe, S., & Schütze, H. (2015). Autoextend: Extending word embeddings to embeddings for synsets and lexemes. *ArXiv Preprint ArXiv:1507.01127*.

Saad, M., & Ashour, W. (2010). OSAC: Open Source Arabic Corpora. *6th International Conference on Electrical and Computer Systems (EECS'10), Nov 25-26, 2010, Lefke, Cyprus.*, 118–123.

Salton, G. (1968). *Automatic information organization and retrieval*.

Šarić, F., Glavaš, G., Karan, M., Šnajder, J., & Bašić, B. D. (2012). Takelab: Systems for measuring semantic text similarity. ** SEM 2012: The First Joint Conference on Lexical and Computational Semantics–Volume 1: Proceedings of the Main Conference and the Shared Task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, 441–448.

Shaalan, K., Al-Sheikh, S., & Oroumchian, F. (2012). Query expansion based-on similarity of terms for improving Arabic information retrieval. *International Conference on Intelligent Information Processing*, 167–176. Springer.

Silberztein, M., Váradi, T., & Tadić, M. (2012). Open source multi-platform NooJ for NLP. *Proceedings of COLING 2012: Demonstration Papers*, 401–408.

Stairmand, M. A. (1997). Textual context analysis for information retrieval. *SIGIR Forum (ACM Special Interest Group on Information Retrieval)*, *31*(1 SPEC. ISS.), 140–147. https://doi.org/10.1145/278459.258552

Su, X., & Gulla, J. A. (2006). An information retrieval approach to ontology mapping. *Data & Knowledge Engineering*, *58*(1), 47–69.

Taghva, K., Elkhoury, R., & Coombs, J. (2005). Arabic stemming without a root dictionary. *International Conference on Information Technology: Coding and Computing (ITCC'05)-Volume II*, *1*, 152–157. IEEE.

Tonon, A., Demartini, G., & Cudré-Mauroux, P. (2015). Pooling-based continuous evaluation of information retrieval systems. *Information Retrieval*, *18*(5), 445–472. https://doi.org/10.1007/s10791-015-9266-y

Tounsi, L., Attia, M., & van Genabith, J. (2009). *Parsing Arabic using treebank-based LFG resources*.

Uzuner, O., Katz, B., & Yuret, D. (1999). Word sense disambiguation for information retrieval. *AAAI/IAAI*, *985*.

Voorhees, E. M. (1993). Using WordNet to disambiguate word senses for text retrieval. *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 171–180.

Voorhees, E. M., & Harman, D. K. (2005). *TREC: Experiment and evaluation in information retrieval* (Vol. 63). MIT press Cambridge.

Wei, X., & Croft, W. B. (2006). LDA-based document models for ad-hoc retrieval. *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 178–185.

Wu, Z., & Palmer, M. (1994). Verb semantics and lexical selection. *ArXiv Preprint Cmp-Lg/9406033*.

Xu, J., & Croft, W. B. (1996). *Query expansion using local and global document analysis In: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*.

Xu, J., & Croft, W. B. (2000). Improving the effectiveness of information retrieval with local context analysis. *ACM Transactions on Information Systems (TOIS)*, *18*(1), 79–112.

Xu, J., Fraser, A., & Weischedel, R. (2002). Empirical studies in strategies for Arabic retrieval. *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 269–274.

Yang, Z., Zhao, J., Dhingra, B., He, K., Cohen, W. W., Salakhutdinov, R., & LeCun, Y. (2018). Glomo: Unsupervisedly learned relational graphs as transferable representations. *ArXiv Preprint ArXiv:1806.05662*.

Yi, X., & Allan, J. (2009). A comparative study of utilizing topic models for information retrieval. *European Conference on Information Retrieval*, 29–41. Springer.