MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE
LA RECHERCHE SCIENTIFIQUE
UNIVERSITE ABDELHAMID IBN BADIS - MOSTAGANEM

Faculté des Sciences Exactes et de l'Informatique
Département de Mathématiques et d'Informatique
Filière : Informatique

MEMOIRE DE FIN D'ETUDES
Pour l'Obtention du Diplôme de Master en Informatique
Option : **Ingénierie des Systèmes d'Information**

THEME :

Identification Automatique d'Entités Nommées

Etudiante : Flitti Sarah

Encadrante : Kenniche Ahlem

Année Universitaire 2016/2017

Résume

Ces dernières décennies, le développement considérable des technologies de l'information et de la communication a modifié en profondeur la manière dont nous avons accès aux connaissances. Face à l'afflux de données et à leur diversité, il est nécessaire de mettre au point des technologies performantes et robustes pour y rechercher des informations, Le web est maintenant devenu une importante source d'information et de divertissement pour un grand nombre de personnes et les techniques pour accéder au contenu désiré ne cessent d'évoluer. Par exemple, en plus de la liste de pages web connue, certains moteurs de recherche présentent maintenant directement, lorsque possible, l'information recherchée par l'usager. Dans ce contexte, l'étude des requêtes soumises à ce type de moteur de recherche devient un outil pouvant aider à perfectionner ce genre de système et ainsi améliorer l'expérience d'utilisation de ses usagers. Dans cette optique, le but de ce projet et de présenter certaines techniques pour la recherche d'information. Notre principale problématique porte l'identification des entités nommées contenues dans un ensemble de documents et de requêtes qui ont été soumises à un moteur de recherche.
Mots-clés : recherche d'information, identification d'entités, indexation, lucene.

Abstract

These last decades, the considerable development of the technologies of information and the communication modified in depth the manner of which we have access to the knowledge. Facing the influx of data and to their diversity, it is necessary to finalize of the effective and robust technologies to search for some information there, The web now became an important source of information and entertainment for a big number of people and the techniques to reach the wanted content quit to evolve. for example, in addition to the list of pages web known, some search engines now present directly, when possible, information searched for by the user. In this context, the survey of the requests submitted to this type of search engine becomes a tool capable to help to perfect this kind of system and so to improve the experience of use of his/her/its users. In this optics, the goal of this project and to present some techniques that have been developed to make the survey of the research web requests submitted to a search engine. Our problematic principal carries l on the detection non supervised of the named entities contained in a set of requests that has been submitted to a search engine.

Keywords: research of information, non supervised classification, identification of entities, indexation.

Dédicaces

Avant tout, je tien à remercies le bon dieu, et l'unique qui m'offre le courage et la volonté nécessaire pour affronter les différentes de la vie, Je dédie ce modeste travail.

À mes très chers parents, Que Dieu les gardent.

À mon cher marie « Brahim ».

À tous mes très chers frères : Abd Elmalek, Oussama, Yassine.

À ma très chères sœurs : Kheira, et Nadjoua.

À ma très chère amie : Asmâa Laredj.

À toute mes amies : Ahlem, Liela, Hafsa, Wassila, Khadidja, Meriem Et Malika.

À tous ceux qui sont proches de mon cœur.

Et dont je n'ai pas cité les noms.

Je dédie ce modeste travail.

Sarah

Remerciements

Tout d'abord au ***Grand Dieu*** *;*

*A notre encadreur M^me^* ***Ahlem Kenniche*** *pour nous avoir fait partager ses connaissances et les lignes directives qu'elle a apporté à ce travail.*

J'adresse mes plus sincères remerciements aux membres du jury d'avoir accepté d'examiner ce modeste travail.

Mes sincères dévouements et profondes gratitude à mes parents, frères et sœurs et mon marie pour leurs encouragement et soutient durant tout le long de mon cycle d'études.

On adresse nos profondes reconnaissance à

MR Kader Hamadi qui contribué à notre formation.

Je tiens de remercier aussi ma très chère amie Mlle ***Asma Laredj*** *qui m'a vraiment aidé afin de réaliser ce modeste travail.*

Nous remerciement tous ceux qui nous ont soutenus de loin ou de prés durant notre cycle d'étude.

Sarah

Sommaire

Chapitre I: La recherche d'information 4
I.1. Introduction 4
I.2. La définition 4
I.3. Recherche dans le web 4
I.4. Concepts de base de la recherche d'information 4
I.4.1.Collection de documents 4
I.4.2. Document 4
I.4.3. Besoin d'information 5
I.4.4. Requête 5
I.4.5. Modèle de représentation 5
I.4.6. Modèle de recherche 5
I.5. Le Système de Recherche d'Information 5
I.6. Le processus de recherche d'information 6
I.6.1. L'indexation 7
I.7. Les modèles de recherche d'information 8
I.7.1. Modèles ensemblistes 8
I.7.2. Modèles algébriques 8
I.7.3. Modèles probabilistes 9
I.8. Outils de recherche d'information 9
I.8.1. Moteurs de recherche 9
I.8.2. Annuaires de recherche 9
I.8.3. Les méta-moteurs 9
I.9.L'extraction d'informations 9
I.9.1. Principe de l'extraction d'information 9
I.10. Annotation sémantique 10
I.11. Conclusion 10
Chapitre II : Etat de l'art sur l'Entités Nommées 12
II.1. Introduction 12
II.2. La Définition 12
II.3. Les formes des entités nommées 12
II.4. Rôle de l'entité nommée 13
II.5. Reconnaissance des entités nommées 13
II.5.1. Approches orientées connaissances 13
II.5.2. Approches orientées données 13

II.5.3. Approches hybrides 14
II.6. Catégorisation 15
II.7. Détection d'entités nommées 15
II.8. Approches d'extraction d'entités nommées 15
II.9. Annotation et évaluation des entités nommées 16
II.9.1. Annotation manuelle de corpus 16
II.9.2. Métriques d'évaluation des entités nommées 17
II.10. Les différents types de systèmes 17
II.11. Quelques problématiques liées aux entités nommées 17
II.12. Campagnes d'évaluation des entités nommées 18
II.13. Conclusion 19
Chapitre III : Conception de notre système d'identification automatique d'entités nommées 21
III.1. Introduction 21
III.2. Processus de recherche d'informations 21
III.3. Le moteur de recherche Lucene 21
III.3.1. Architecture et fonctionnement de Lucene 22
III.4.Term Frequency-Inverse Document Frequency 23
III.4.1.Tf (termfrequency) : (pondération locale). 24
III.4.2. Idf (Inverse of Document Frequency): (pondération globale) 24
III.5. Indexation des données 26
III.5.1. Indexation avec lucene 26
III.5.2. Indexation de textes 27
III.6. L'annotation 30
III.6.1. L'annotation d'entités nommées 30
III.6.2. Stanford Nommée Entité Recognizer (NER) 30
III.6.3. Utilisation programmée via l'API 31
III.6.4. Utilisation programmée via un service 31
III.6.5. Les modèles 31
III.7. La Recherche 32
III.8. Classes 34
III.8.1. Classes d'indexation 34
Chapitre IV : Implémentation et mise œuvre 38
IV.1. Introduction : 38
IV.2. Environnement de travail : 38
IV.2.1. Ressources utilisées : 38
IV.2.2. Pourquoi JAVA : 38

IV.2.3. Pourquoi NetBeans ?.....39
IV.3. Architecture de L'application.....39
IV.4. Corpus de teste.....40
IV.5. Présentation de l'application :.....41
IV.6. Menu principale.....42
IV.6.1. Corpus.....42
IV.6.2. Recherche par entité.....43
IV.6.3. Evaluation.....47
IV.7. Conclusion.....48

Liste des tableaux

Table I.1. Exemples d'entités nommées……..…...9

Table II.2. Les résultats de certains systèmes de REN sur le corpus MUC- 7 en termes de F-mesure. ……..14

Tableau II.3. Caractéristiques des principales campagnes d'évaluation…… ..……..18

Tableau III.1. Les Analyseur fournis par Lucene. …….…….34

Tableau III.2. Détail des métadonnées de Field. ……...…….……35

Liste des algorithmes

Algorithme 3.1. vectorisation des fichiers…….……..28

Algorithme 3.2. Racinisation et Lemmatisation.……...…….29

Algorithme 3.3. TF-IDF (termfrequency - inverse document frequency)..……..…….29

Algorithme 3.4. La recherche……...…. 33

Liste des figures

Figure I.1. Architecture générale d'un Système de Recherche d'Information.……6

Figure I.2. Processus en U de recherche d'informations…….…….…7

Figure I.3. Représentation du modèle vectoriel.....8

Figure I.4. Plate-forme d'annotation sémantique.....10

Figure II.1. Processus de la classification supervisée ……...…….…….14

Figure II.2. Éléments d'un processus d'annotation……..…...16

Figure III.1. Indexation d'un document. [22] …….……21

Figure III.2. Processus d'indexation de Lucene. [20]……..…….…… 22

Figure III.3. Architecture générale de Lucene [20]…….....23

Figure III.4. Architecture d'indexation [23]…….... 27

Figure III.5. Fonctionnent de NER…… 31

Figure III.6. Architecture et organisation de Lucene [23]…….. 32

Figure III.7. Recherche par type annoté…… 33

Figure IV.1. Architecture de L'application…….…….39

Figure IV.2. Exemple du corpus Reuters……... 41

Figure IV.3. L'interface principale de notre application…….……42

Figure IV.4. Barre d'outille de notre application…… 42

Figure IV.5. Interface de consultation du corpus……... 43

Figure IV.6. Fenêtre d'indexation du corpus…….. 44

Figure IV.7. Fenêtre de l'annotation du corpus……... 45

Figure IV.8. Fenêtre de l'index entités…….. 46

Figure IV.9. L'interface de recherche…….…… 47

Figure IV.10. L'interface d'évaluation……...…… 48

Liste d'abréviations

RI : Recherche d'Information.

SRI: Système de Recherche d'Information.

EI : Extraction d'Information.

EN : Entités Nommées.

RE : Recherche d'Entités.

REN : Reconnaissance des Entités Nommées.

EN-PERS : Entité Nommée de type nom de Personne.

EN-LOC : Entité Nommée de type nom de Lieu.

EN-ORG : Entité Nommée de type nom d'Organisation.

TAL : Traitement Automatiques des Langues.

TALN : Traitement Automatique du Langage Naturel.

TA : Traduction Automatique.

WIE : Web Information Extraction.

MUC : Message Understanding Conference.

MHV : Mots Hors Vocabulaire.

SVM : Les Machines à Vecteurs de Support.

HMM : Le Modèle de Markov à états cachés.

NERC : Named Entity Recognition and Classification.

MMC : Le Modèle de Markov Caché.

MEM : Maximum Entropy Model.

CRF : Conditional Random Field.

FST : Transducteurs à Etats Finis.

LTG : Language Technology Group.

HTTP : HyperText Transfer Protocol.

CoLL : Conference on Natural Language Learning.

QR : Système de Question Réponses.

ACE : Automatic Content Extraction.

ESTER : Evaluation des Systèmes de Transcription Enrichie d'Emissions Radiophonique.

JMF : Java Media Framework.

EDI : Environnement de Développement Intégré.

TAS : Typed Annotated Search.

Introduction générale

La recherche d'information est matérialisée par un ensemble d'outils dont l'objectif est de répondre à un besoin applicatif à partir d'une collection de données d'une manière automatique. On parle alors d'extraction d'information, L'extraction d'information (El) qui est un sous-domaine du traitement automatique du langage naturel (TALN) consiste à extraire automatiquement, à partir de données non (ou semi) structurées, des informations structurées pertinentes pour une tâche particulière. Dans ce rapport, nous nous intéressons à l'une des sous-tâches de l'El qui est la reconnaissance et la détection des entités nommées (ENs).

La détection des Entités Nommées (EN) est un élément essentiel à de nombreuses tâches de TAL, comme la recherche d'information ou la traduction automatique. En témoignent les nombreuses campagnes d'évaluation internationales (MUC, CoNLL, ACE) ou nationales (ESTER) organisées au cours des 15 dernières années. Les entités nommées (EN), est une appellation générique pour les noms propres désignant entre autres des personnes, des lieux ou des organisations. Comme la plupart des unités lexicales considérées en dehors du contexte d'un énoncé, les EN sont polysémiques.

La reconnaissance des entités nommées est centrale dans bon nombre de problématiques en recherche d'information comme par exemple :

Questions-Réponses (QR). Cette tâche a connu un fort engouement ces dernières années. Un système QR présente au moins deux différences par rapport à un système de recherche d'information(RI). La première est la formulation de la requête qui est une phrase interrogative en langage naturel (par exemple "Je veux connaître les spécifications techniques du nouveau Blackberry"). Cela a de l'intérêt pour les utilisateurs (la formulation de requêtes efficaces sous forme de mots clés est une tâche difficile) et pour les systèmes (apport d'un contexte et d'informations supplémentaires). La seconde principale différence est la forme des résultats : un moteur de RI va retourner une liste de documents, dans lesquels l'utilisateur va être en charge de trouver la réponse par lui-même, tandis qu'en QR, le système doit retourner une série de réponses précises (c'est à- dire des chaînes correspondant exactement à ce que l'utilisateur recherche), généralement des entités nommées. C'est pourquoi une identification correcte des entités nommées est une étape vitale.

L'objectif de notre travail est de développer un processus d'extraction d'information pour l'identification automatique d'entités nommées.

Ce rapport est structuré comme suit :

- Le premier chapitre appelé **la Recherche d'information** qui décrit le problème de la Recherche d'information. Tel que on présente les concepts de base de la RI, les différents modèles pour effectuer ce traitement et décrit en générale le processus de la RI.

- Le deuxième chapitre que nous avons appelée **Etat de l'art sur les Entités Nommées**, dans lequel nous présentons les entités nommées, ces concepts clés et les techniques d'indentifications automatiques.

- Le troisième chapitre appelé **conception de notre système d'identification automatique d'entités,** présente les différents algorithmes que nous avons appliqué pour l'implémentation de notre système et nous abordons les aspects de conception de notre solution

- Le quatrième chapitre appelé **implémentation de système de d'identification automatique d'entités** expose toute les interfaces, les fonctionnalités et les différentes étapes d'implémentation et d'expérimentation de notre système d'identification automatique d'entités**.**

Nous terminons ce mémoire par une conclusion générale.

CHAPITRE I :

Recherche d'Information

Chapitre I: La recherche d'information

I.1. Introduction

La recherche d'information est une branche de l'informatique qui s'intéresse à l'acquisition, l'organisation, le stockage, la recherche et la sélection d'information, L'opération de la RI est réalisée par des outils informatiques appelés Systèmes de Recherche d'Information (SRI), Cette partie a pour but de présenter le domaine de la Recherche d'information (RI). On présente les concepts de base de la RI. On décrit tout d'abord la structure générale du système de recherche d'information, et le processus général de RI, pour retrouver parmi un ensemble de documents ceux qui répondent précisément à la requête d'un utilisateur. Nous décrivons ensuite la notion de processus d'indexation, et les diverses méthodes de l'indexation. Nous aborderons un bref aperçu des différents modèles de recherche d'information .Enfin, nous étudierons la notion et le principe de l'extraction d'information.

I.2. La définition

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. [1]

I.3. Recherche dans le web

L'objectif de la recherche d'information dans le web est de satisfaire les besoins des utilisateurs en information.

Les utilisateurs éffectuent leurs recherches de plusieurs manières: de manière navigationelle (me donner l'url du site que je veux atteindre), transactionnelle (me montrer des sites où je peux éffectuer une transaction, par exemple, télécharger un fichier ou trouver une carte) ou informationnelle (chercher des informations dans plusieurs pages web). [3]

I.4. Concepts de base de la recherche d'information

La recherche d'information est considérée comme l'ensemble des techniques permettant de sélectionner à partir d'une collection de documents, ceux qui sont susceptibles de répondre aux besoins de l'utilisateur. La gestion de ces informations implique le stockage, la recherche et l'exploration des documents pertinents. De ce contexte plusieurs concepts clés peuvent être définis, on a donc trouvé utile de les clarifier [1].On a permis de dégager les concepts suivants:

I.4.1.Collection de documents

La collection de documents constitue l'ensemble des informations exploitables et accessibles. Elle est constituée d'un ensemble de documents. Dans le cas général et pour un souci d'optimalité, la base constitue des représentations simplifiées mais suffisantes pour ces documents. Ces représentations sont étudiées de telles sortes que la gestion (ajout suppression d'un document) ou l'interrogation (recherche) de la base se font dans les meilleures conditions de coût.

I.4.2. Document: Le document constitue l'information élémentaire d'une collection de documents. L'information élémentaire, appelée aussi granule de document, peut représenter tout ou une partie d'un document. [1]

I.4.3. Besoin d'information

La notion de besoin en information en recherche d'informations est souvent assimilée au besoin de l'utilisateur. Trois types de besoin utilisateur ont été définis par :

Besoin vérificatif : l'utilisateur cherche à vérifier le texte avec les données connues qu'il possède déjà. Il recherche donc une donnée particulière, et sait même souvent comment y accéder. La recherche d'un article sur Internet à partir d'une adresse connue serait un exemple d'un tel besoin. Un autre exemple serait de chercher la date de publication d'un ouvrage dont la référence est connue. Un besoin de type vérificatif est dit stable, c'est-à-dire qu'il ne change pas au cours de la recherche. [1]

Besoin thématique connu : l'utilisateur cherche à clarifier, à revoir ou à trouver de nouvelles informations dans un sujet et un domaine connus. Un besoin de ce type peut être stable ou variable : il est très possible en effet que le besoin de l'utilisateur s'affine au cours de la recherche. Le besoin peut aussi s'exprimer de façon incomplète, c'est-à dire que l'utilisateur n'énonce pas nécessairement tout ce qu'il sait dans sa requête mais seulement un sous-ensemble.

Besoin thématique inconnu : cette fois, l'utilisateur cherche de nouveaux concepts ou de nouvelles relations en dehors des sujets ou des domaines qui lui sont familiers. Le besoin est intrinsèquement variable et est toujours exprimé de façon incomplète. [1]

I.4.4. Requête

La requête constitue l'expression du besoin en information de l'utilisateur. Elle représente l'interface entre le SRI et l'utilisateur. Une requête est un ensemble de mots clés, mais elle peut être exprimée en langage naturel, booléen ou graphique.

I.4.5. Modèle de représentation

Un modèle de représentation est un processus permettant d'extraire d'un document ou d'une requête, une représentation paramétrée qui couvre au mieux son contenu sémantique. Ce processus de conversion est appelé indexation. Le résultat de l'indexation constitue le descripteur du document ou de la requête, qui est une liste de termes ou groupes de termes (concepts), significatifs pour l'unité textuelle correspondante.

I.4.6. Modèle de recherche

Il représente le modèle du noyau d'un SRI. Il comprend la fonction de décision fondamentale qui permet d'associer à une requête, l'ensemble des documents pertinents à restituer.

I.5. Le Système de Recherche d'Information

Un SRI est un système informatique qui permet de retourner à partir d'un ensemble de documents, ceux dont le contenu correspond le mieux à un besoin en informations d'un utilisateur, exprimé à l'aide d'une requête.

L'architecture générale d'un SRI illustrée par la figure I.1

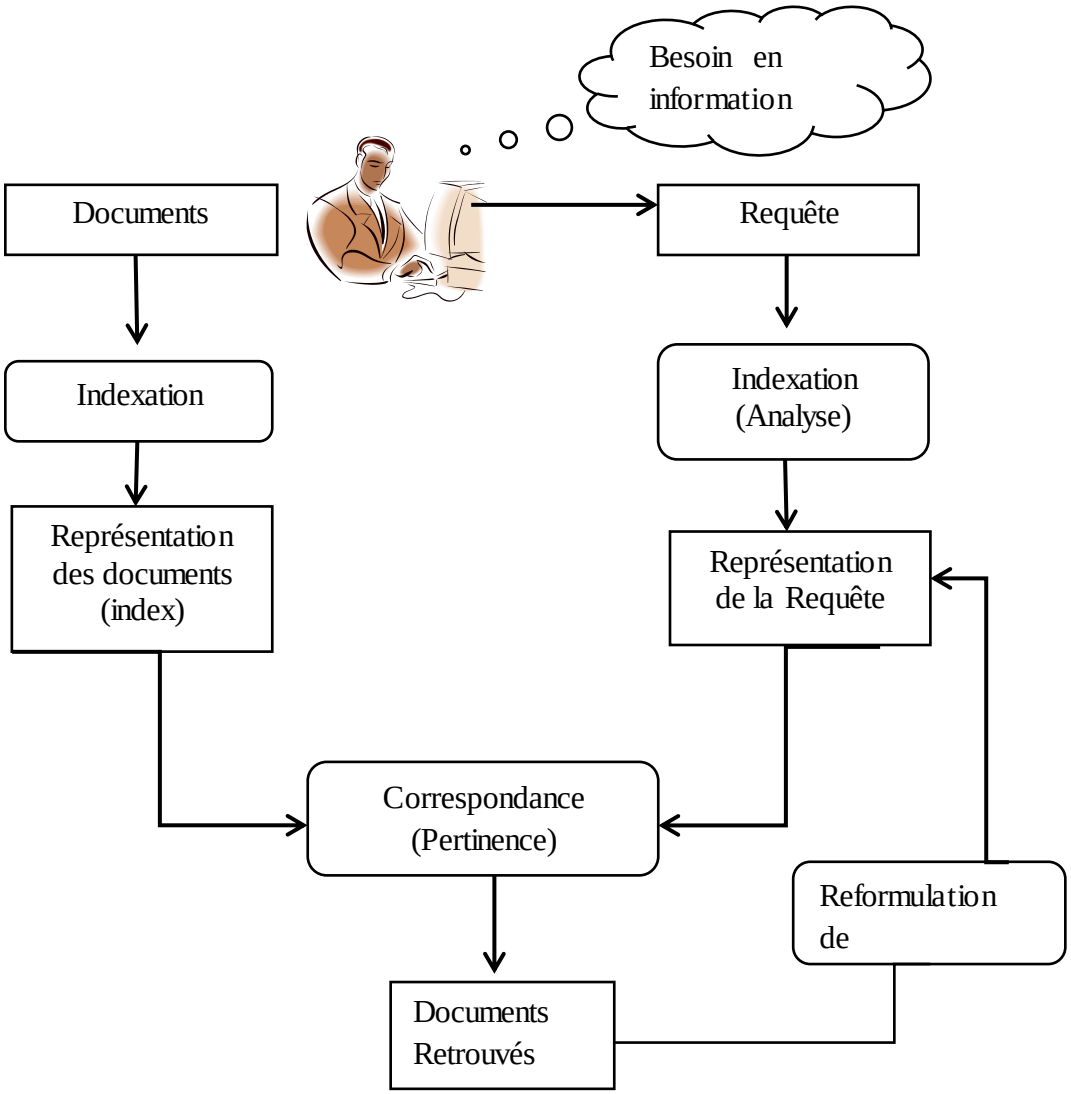


Figure I.1. Architecture générale d'un Système de Recherche d'Information. [6]

I.6. Le processus de recherche d'information

Les différentes étapes du processus de RI, sont représentées schématiquement par le processus en U (voir FigureI.2). La figure illustre particulièrement :

- les notions de documents et de requêtes qui sont des conteneurs d'informations.
- les opérations d'analyse, d'indexation et d'appariement qui permettent globalement de traiter la requête dans le but de sélectionner des documents à présenter à l'utilisateur.

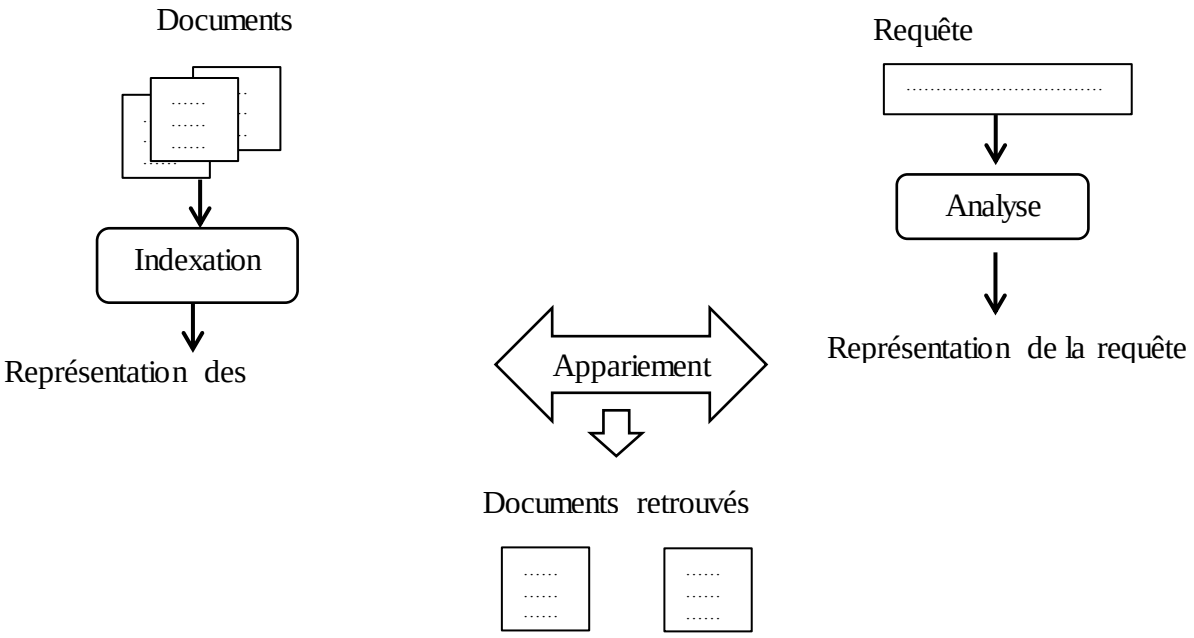


Figure I.2. Processus en U de recherche d'informations.

Document : Le document représente le conteneur élémentaire d'information, exploitable et accessible par le SRI. Un document peut être un texte, une page WEB, une image, une bande vidéo, etc.

Requête : Une requête constitue l'expression du besoin en informations de l'utilisateur.

L'appariement requête-document : Les SRI intègrent un processus de recherche/décision qui permet de sélectionner l'information jugée pertinente pour l'utilisateur.

L'objectif fondamental d'un processus de RI est de sélectionner les documents ''les plus proches'' du besoin en information de l'utilisateur décrit par une requête. [1]

I.6.1. L'indexation

Un SRI gère les différentes collections de documents en les organisant sous forme d'une représentation intermédiaire permettant de refléter aussi fidèlement que possible leur contenu sémantique, L'indexation est une étape très importante dans le processus de RI. [1]

Le processus l'indexation : Pour que la recherche d'information se réalise avec des coûts acceptables, il convient d'effectuer une opération fondamentale sur les documents de la collection. Cette opération est nommée indexation [6]. Finalité de l'indexation est donc de produire une représentation synthétique des documents, formé de termes, ces termes peuvent être extraits de trois manières :

Manuelle : Chaque document de la collection est analysé par un spécialiste du domaine ou un documentaliste. L'indexation manuelle assure une meilleure précision dans les documents restitués par le SRI en réponse aux requêtes des utilisateurs. [6]

Semi-automatique : La tâche d'indexation est réalisée ici simultanément par un programme informatique et un spécialiste du domaine [6], le choix final reste au spécialiste du domaine correspondant ou documentaliste, qui intervient souvent pour établir des relations sémantiques entre mots-clés et choisir les termes significatifs. [1]

Automatique : Chaque document est analysé à l'aide d'un processus entièrement automatisé [1], Elle est réalisée par un programme informatique et elle passe par un ensemble d'étapes pour créer d'une façon automatique l'index. Ces étapes sont: l'analyse lexicale, l'élimination des mots vides, la normalisation (lemmatisation), la sélection des descripteurs, le calcul de

statistiques sur les descripteurs et les documents (fréquence d'apparition d'un descripteur dans un document et dans la collection, la taille de chaque document, etc.) et enfin la création de l'index et éventuellement sa compression. [6]

I.7. Les modèles de recherche d'information

De nombreux modèles ont été proposés en RI, ils sont généralement regroupés autour des trois familles suivantes:

• les modèles ensemblistes qui considèrent le processus de recherche comme une succession d'opérations à effectuer sur des ensembles d'unités lexicales contenues dans les documents,

• les modèles algébriques au sein desquels la pertinence d'un document par rapport à une requête est envisagée à partir de mesures de distance dans un espace vectoriel.

• les modèles probabilistes qui représentent la RI comme un processus incertain et imprécis où la notion de pertinence peut être vue comme une probabilité de pertinence. [17]

I.7.1. Modèles ensemblistes

Nous nous intéressons ici uniquement au principal représentant des modèles inspirés de la logique booléenne et de la théorie des ensembles pour modéliser l'appariement entre une requête et les documents de la collection : le modèle booléen classique, Le modèle booléen est le modèle le plus ancien et également le plus simple en RI. [17]

Un document est représenté par un ensemble de mots-clés (termes) ou encore un vecteur booléen. La requête de l'utilisateur est représentée par une expression logique. [6]

I.7.2. Modèles algébriques

Les modèles algébriques considèrent les documents et les requêtes comme faisant partie d'un même espace vectoriel, et leur appariement est fait suivant une mesure algébrique de similarité. Parmi les différentes variantes de ce type de modèle, le plus connu est le modèle vectoriel. [17]

Modèle vectoriel : Ce modèle se base sur une formalisation géométrique. En effet, les documents et les requêtes sont représentés dans un même espace, défini par un ensemble de dimensions, chaque dimension représente un terme d'indexation. [6]

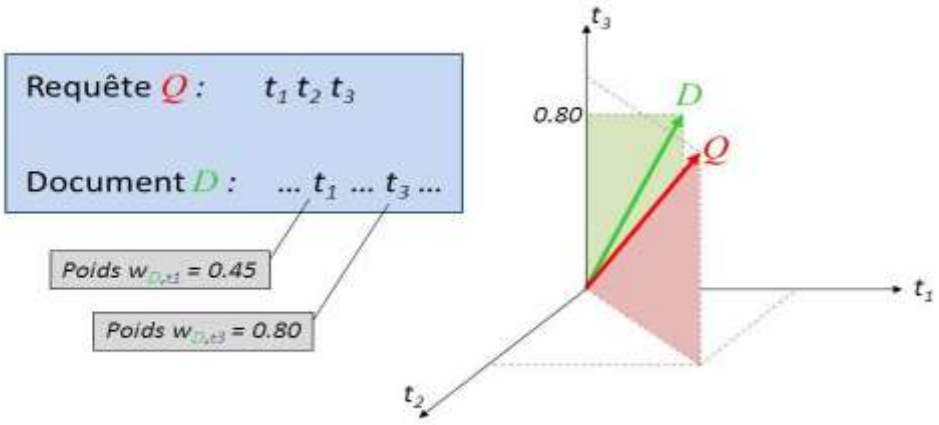


Figure I.3. Représentation du modèle vectoriel

I.7.3. Modèles probabilistes

Le modèle probabiliste représente la probabilité de la pertinence d'un document D par rapport à une requête R. Le but de cette fonction de similarité dans ce modèle est d'essayer de séparer les documents pertinents des non pertinents au sein d'une collection. L'idée de base, dans ce modèle, est de tenter de déterminer les probabilités P(R/D) et P(NR/D) pour une requête donnée. [17]

I.8. Outils de recherche d'information

Il existe de nombreux outils de recherche d'information sur le Web, Ces outils qui se spécialisent en fonction des services utilisés et du type d'information qu'ils recensent.

I.8.1. Moteurs de recherche

Un moteur de recherche est un site indexant tous les autres sites internet et vous permet de poser une question ou taper des mots pour faire une recherche. Le moteur va ensuite retourner les résultats les plus pertinents. Google est actuellement le moteur de recherche le plus utilisé dans le monde. On retrouve aussi Yahoo! et Bing de Microsoft [18]. Un moteur de recherche est une application permettant de retrouver des ressources (pages web, images, vidéo, fichiers, etc.) associées à des mots quelconques. [1]

I.8.2. Annuaires de recherche

Les annuaires de recherche permettent également des recherches par mot-clé, mais ils sont surtout construits sur le modèle d'une arborescence thématique. Les sites qu'ils répertorient sont visités "manuellement" (et non de façon automatique comme par les robots des moteurs de recherche). Exemple : l'annuaire 'Wohaa', l'annuaire russe 'Yandex'.

I.8.3. Les méta-moteurs

Ce sont des moteurs intégrant plusieurs moteurs et permettant donc de lancer une recherche dans plusieurs directions en même temps. Le plus connu est Copernic

I.9.L'extraction d'informations

La création de systèmes d'extraction d'information est devenue de plus en plus ombreux depuis quelques années. Toutefois, plusieurs défis se dressent devant ces systèmes avant qu'ils atteignent des performances optimales, En effet, il faut tenir compte du fait que les données textuelles contiennent souvent de l'information non structurée, ce qui rend l'extraction d'information plus complexe. En plus du développement des systèmes d'extraction d'information, les travaux de recherche essayent de répondre à cette problématique de surcharge d'informations en développant d'autres outils spécifiques tels que : les moteurs de recherche, les analyseurs Morphologiques et syntaxiques, …etc. [3]

I.9.1. Principe de l'extraction d'information

L'extraction d'information consiste à analyser (souvent superficiellement) des textes pour en obtenir des informations en vue d'une application précise. L'extraction d'information ne cherche pas à comprendre les textes dans leur ensemble, Elle fait la recherche d'une information spécifique et extrait d'un texte donné des éléments pertinents. Généralement le type d'une information pertinente pour une application donnée est défini à l'avance. [3]

L'extraction d'information dans le web (Web Information Extraction, WIE) sert à identifier et extraire l'information du web systématiquement.

I.10. Annotation sémantique

L'annotation est le processus qui consiste à attacher des informations complémentaires au contenu textuel d'un document. L'annotation sémantique consiste à relier ces contenus à des informations précises (on parle parfois de métadonnée) en relation avec l'identité sémantique des données annotées. De plus en plus fréquemment, on considère la tâche d'annotation sémantique comme l'un des aspects applicatifs du Web sémantique.

Exemple d'application : annotation de document.

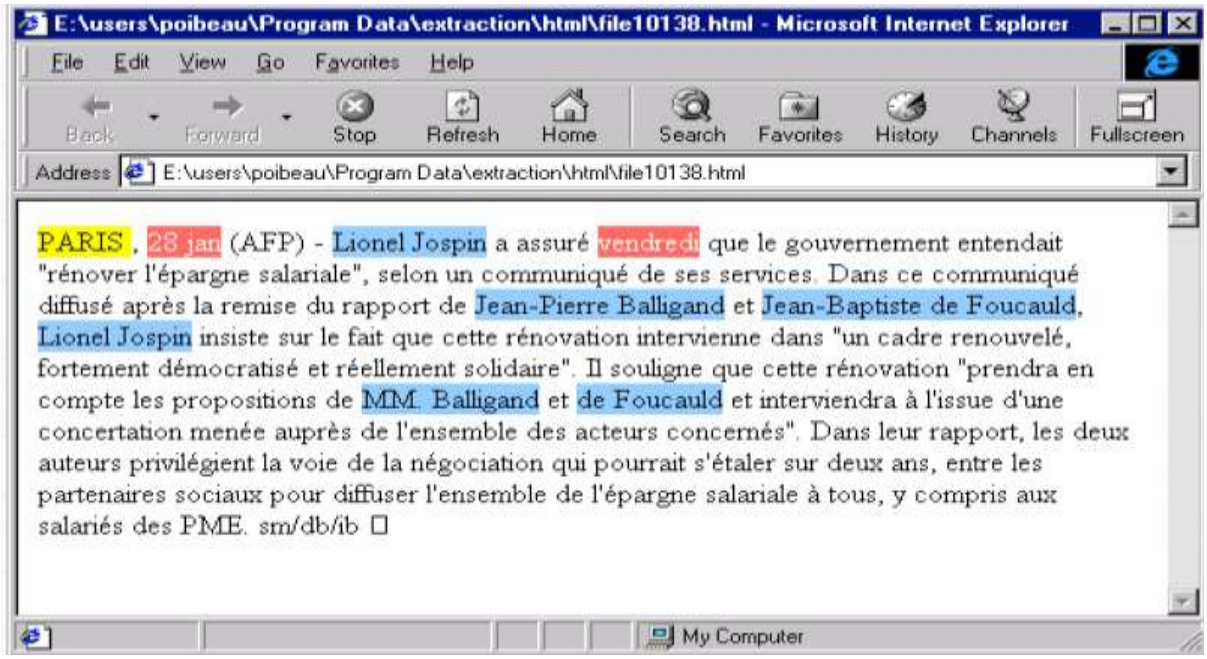


Figure I.4. Plate-forme d'annotation sémantique. [16]

I.11. Conclusion

Dans ce chapitre nous avons présenté les principales notions et concepts de la recherche d'information, des systèmes de recherche d'information et ceux des outils de recherche sur le web. A travers les différentes sections que nous avons présentées, nous concluons que la recherche d'information, s'attache à définir des modèles et des systèmes afin de faciliter l'accès à un ensemble de documents se trouvant dans des bases documentaires ou encore sur le web. Le but est de permettre aux utilisateurs de retrouver les documents dont le contenu répond à leur besoin en information, il s'agit donc de retourner l'ensemble de documents pertinents.

Dans le chapitre suivant, Nous détaillerons encore plus les problématiques relatives à notre thème, à savoir, la recherche d'information, Les entités nommées et leurs identification automatique.

CHAPITRE II :

Les Entités Nommées : état d’art

Chapitre II : Etat de l'art sur l'Entités Nommées

II.1. Introduction

Le concept d'entité nommée est apparu dans les années 90, Les entités nommées constituent un champ de recherche très actif depuis de nombreuses années. Elles sont depuis longtemps considérées comme un point central dans de nombreuses applications mettant en jeu des notions comme la compréhension, la recherche sémantique, etc. La détection des entités nommées (EN) est un élément essentiel pour de nombreuses tâches de traitement automatique des langues (TAL), qu'elles soient monolingues ou multilingues, Nous allons essayer de toucher brièvement certaines des bases d'entités nommées, et les méthodes pour les détecter.

Dans cette section, nous commençons à définir l'EN ensuite, nous présentons quelques notions de base liées au domaine d'extraction des ENs.

II.2. La Définition

On appelle traditionnellement « entités nommées » (de l'anglais named entity) l'ensemble des noms de personnes, d'entreprises et de lieux présents dans un texte donné. On associe souvent à ces éléments d'autres syntagmes comme les dates, les unités monétaires ou les pourcentages. [11]

Entité nommée	**Type**
Barack Obama	Nom de personne
USA	USA Nom de lieu
29 Décembre 2015	Expression temporelle
UNICEF	Nom d'organisation

Table I.1. Exemples d'entités nommées

II.3. Les formes des entités nommées

Il y a deux formes d'EN ; Les ENs simples et les ENs composées. Chaque forme est traitée différemment.

- ✓ **Les entités nommées simples**

Une EN simple est une EN qui est composée d'un seul mot, comme les noms de lieu « Canada » et « Algérie »ou le nom de personne « Ali ».

- ✓ **Les entités nommées composées**

Une EN composée est une EN qui est composée de deux ou plusieurs mots, comme par exemple le nom de personne « Houari Boumédiène » et le nom de lieu « Afrique du Sud ».

II.4. Rôle de l'entité nommée

Les ENs présentent plusieurs avantages dans le domaine de la recherche en TALN. Elles sont par exemple utiles pour le développement des systèmes de questions/réponses, les résumés automatiques, la recherche d'information, la traduction automatique (TA), le Web sémantique, et la bio-informatique. Les ENs sont utilisées aussi pour la réduction du taux de mots hors vocabulaire (MHV). [9]

II.5. Reconnaissance des entités nommées

La plupart des systèmes de REN utilisent soit des approches orientées connaissances soit des approches orientées données. Les systèmes orientés connaissances sont fondés sur des lexiques (listes de prénom, de pays, etc.) et sur un ensemble de règles de réécriture. D'un autre côté, les systèmes orientés données sont basés sur un modèle appris à partir d'un corpus préalablement annoté. Afin de profiter des avantages de ces deux approches, d'autres systèmes combinent des techniques d'apprentissage automatique et des règles produites manuellement. [13]

La reconnaissance des entités nommées consiste à :

- Identifier des unités lexicales dans un texte ;
- Les catégoriser ;
- Eventuellement, les normaliser. [3]

II.5.1. Approches orientées connaissances

Pour les approches orientées connaissances, les règles d'extraction sont produites manuellement par des experts en se reposant essentiellement sur des descriptions linguistiques, des indices et des dictionnaires de noms propres et de mots déclencheurs. Ces règles prennent la forme de patrons d'extraction permettant de repérer et de classifier les entités nommées. Exemple : le mot déclencheur « Monsieur » précède un mot inconnu commençant par une majuscule, alors le syntagme peut être étiqueté comme un nom de personne.

Les systèmes orientés connaissances permettent d'obtenir de bons résultats sur les textes bien formés. Exemple : le système français Nemesis[1]. [13]

II.5.2. Approches orientées données

Les approches orientées données visent à apprendre les règles d'extraction de manière autonome. L'acquisition de ces règles ainsi que de certaines ressources de connaissances s'effectue à partir d'un corpus de grande taille de manière supervisée.

Le système de Raymond et Fayolle[2] : un système orienté données de REN pour le français. [13]

---

[1] Nemesis : Un système orienté connaissances de REN pour le français, Nemesis est un système qui permet la délimitation et la catégorisation des entités nommées.

[2] Le système de Raymond et Fayolle utilisent différents algorithmes d'apprentissage automatique pour la reconnaissance d'entités nommées dans les transcriptions de la parole.

L'apprentissage supervisé : consiste à apprendre les règles à partir d'un corpus préalablement annoté. La supervision concerne l'intervention humaine, par le biais d'étiquetage d'une base d'exemples, afin de guider le système lors du processus d'apprentissage. Une méthode d'apprentissage est appliquée pour entraîner le système à exploiter les différents traits singularisant les entités nommées. Ensuite le système d'apprentissage généralise le processus afin de produire un modèle permettant d'extraire les entités nommées dans de nouveaux documents

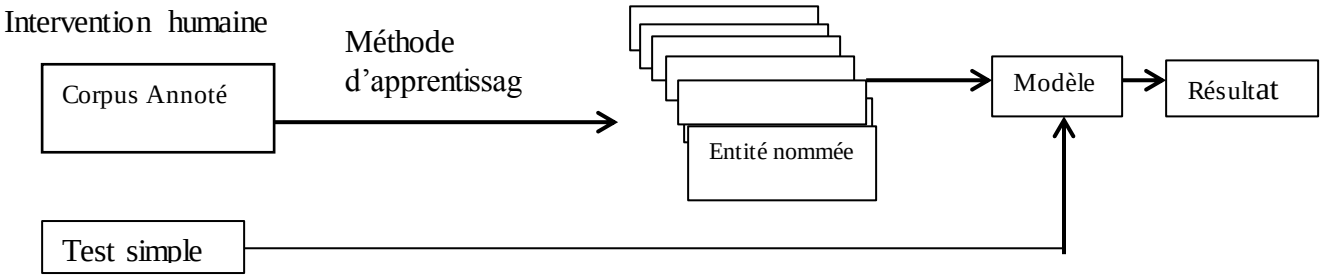


Figure II.1. Processus de la classification supervisée

La performance des systèmes orientés données augmente proportionnellement avec la quantité et la qualité du corpus d'apprentissage.

Les différents systèmes de ce type se basent notamment sur les méthodes d'apprentissage suivantes : machines à vecteurs de support (SVM), modèle de Markov à états cachés (HMM), modèle de l'entropie maximale (EM), modèle de champs conditionnels aléatoires (CRF) et arbres de décision.

II.5.3. Approches hybrides

Certains systèmes tirent profit des avantages respectifs des méthodes orientées connaissances et celles orientées données. Les règles sont, soient apprises automatiquement puis révisées manuellement soient écrites manuellement puis corrigées et améliorées automatiquement. Les systèmes hybrides utilisent conjointement des techniques orientées connaissances et des techniques orientées données. Un ensemble de règles est écrites par un expert puis enrichi automatiquement en utilisant des techniques d'apprentissage, ce qui permet d'obtenir progressivement une meilleure couverture. Exemple : LTG[3] : un système hybride de REN pour l'anglais. [13]

Exemple :

Système	Approche	F-mesure (%)
LTG	Hybride	93,39
MENE	Orienté données (supervisé)	92,20
IsoQuest	orienté connaissances	91,60
BALIE	orienté données (semi-supervisé)	77,71

Table II.2. Les résultats de certains systèmes de REN sur le corpus MUC- *7* en termes de F-mesure. [13]

[3] LTG : Un système hybride a obtenu les meilleurs résultats lors de cette compétition.

II.6. Catégorisation

Par la richesse des informations qu'elles contiennent, les entités nommées (EN) sont des éléments très importants pour les systèmes d'extraction d'information. Dans les années 1980, les campagnes d'évaluation MUC ont permis de définir ce qu'était une tâche de reconnaissance des entités nommées. Pour MUC-6, les EN sont les noms propres, les acronymes et éventuellement d'autres mots qui rentrent dans les catégories suivantes :

- ✓ Organisation : regroupe les entreprises, les institutions gouvernementales et les autres organisations.
- ✓ Person : regroupe les noms de personnes ou de familles.
- ✓ Location : regroupe les noms de lieux politiquement ou géographiquement définis (villes, pays, régions, etc.).
- ✓ Time : regroupe les dates et données temporelles.
- ✓ Nombre : regroupe les données numériques comme les sommes d'argent ou les pourcentages.

II.7. Détection d'entités nommées

Certains moteurs de recherche, qui sont ici référés par le terme MRI, tentent maintenant de fournir directement, pour les requêtes auxquelles ce genre de procédé s'applique, l'information demandée par l'usager par le biais de sa requête. Pour être capable de bien répondre à ce besoin, on suppose que ce genre de système doit, entre autres, être capable de découvrir les entités nommées contenues dans la requête émise par l'usager et d'en identifier leurs types. Cette tâche est habituellement référée par le terme Named Entity Recognition and Classification (NERC). En général, les types d'entités nommées à découvrir et à classifier diffèrent d'un problème à l'autre selon les besoins du système. Dans le contexte de la recherche web, on pourrait vouloir d'un système de NERC qu'il soit capable d'identifier des mentions d'entités nommées telles que, par exemple, des noms d'artistes, d'albums, de chansons, d'athlètes, de restaurants, de compagnies diverses ou de villes et villages.

De façon générale, deux grands types de technique peuvent être utilisés pour la conception d'un système de NERC. Le premier type regroupe les techniques basées sur des ensembles de règles grammaticales et syntaxiques qui ont été construites manuellement pour chaque type d'entité nommée considéré. Le deuxième type regroupe les techniques basées sur des modelés statistiques (Modelé de Markov Cache (MMC), Maximum Entropy Model (MEM) ou encore Conditional Random Field (CRF)) qui seront entraines avec un ensemble de textes dans lesquels les entités nommées à détecter ont déjà été identifiées et classées. Apres avoir été entraines sur ces données, les modèles statistiques peuvent être utilisés pour identifier et classifier les entités nommées présentes dans un segment de texte donne. Il existe également des systèmes hybrides qui utilisent à la fois un ensemble de règles grammaticales et syntaxiques et un ou des modelé(s) statistique(s) pour effectuer cette tâche. [5]

II.8. Approches d'extraction d'entités nommées

Le service d'extraction des entités détecte les références à des personnes, lieux, entreprises, dates, etc. qui sont contenues dans un texte. La REN constitue un champ de recherche très actif depuis de nombreuses années dans plusieurs langues. Des approches fondamentales existent : l'extraction fondée sur des démarches linguistiques ou encore

nommées symboliques, des approches statistiques ou à base d'apprentissage, des approches hybrides qui combinent les deux précédentes. [12]

Approche symbolique : Appelée aussi approche linguistique ou approche à base de règles, elle est utilisée par la majorité des systèmes de reconnaissance d'entités nommées. Son principe de base est d'utiliser des connaissances linguistiques pour établir une liste de règles d'annotation. Ces règles sont écrites manuellement par des experts du domaine (en anglais handcrafted rules), elles portent soit sur les constituants de l'entité nommée, soit sur leur contexte (preuve interne et preuve externe). Elles peuvent être de natures différentes : syntaxiques, informations lexicales, morphologiques ou encore sémantiques. [19]

Approche statistique : Contrairement aux approches symboliques qui reposent sur l'intuition humaine, l'approche statistique appelée aussi approche par apprentissage, utilise des processus automatiques pour l'extraction d'information. Son principe est de mettre au point, d'une manière automatique, des modèles d'analyse à partir de masses importantes de données. [19]

Les techniques mises en œuvre sont basées sur des méthodes hybrides combinant approches symboliques et approches statistiques (apprentissage automatique) Elles permettent de comprendre que dans une phrase comme « Orange n'est pas cotée en bourse », « Orange » réfère à une entreprise, alors que dans « Notre voyage à Orange s'est bien terminé », « Orange » réfère à la ville et que dans « J'ai fait de la confiture à l'orange », « Orange » réfère au fruit et non pas à une entité nommée comme dans les deux précédents. [15]

II.9. Annotation et évaluation des entités nommées

II.9.1. Annotation manuelle de corpus

L'annotation de corpus est une thématique très active qui fait l'objet de nombreux travaux. Effectivement, celle-ci peut être plus ou moins assistée, guidée, automatisée. De plus, comme le montre, Le travail nécessite une grande rigueur et beaucoup de préparation afin d'obtenir une annotation fiable. Dans l'essentiel, trois éléments paraissent indispensables : [10]

- **Guide d'annotation** : Détaille les expressions linguistiques à annoter, selon des critères qui doivent laisser aussi peu de latitude que possible à la personne qui réalisera l'annotation. [10]
- **Outils d'annotation** : Logiciels servant à annoter, dont les interfaces doivent faciliter, mais sans biaiser, le travail de l'annotateur, en incluant éventuellement une phase de pré-annotation automatique. [10]
- **Mesures d'évaluation de la qualité des annotations** : Tests prévus afin de confirmer la fiabilité d'une annotation (accord inter-annotateurs) sur les parties annotées par plusieurs personnes (annotation croisée)

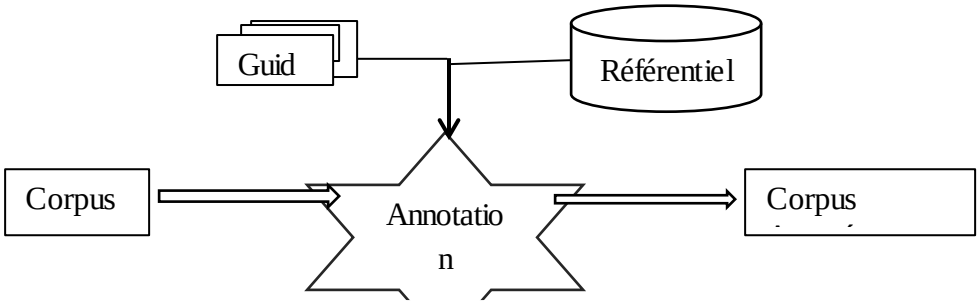


Figure II.2. Éléments d'un processus d'annotation

II.9.2. Métriques d'évaluation des entités nommées

Le rappel, la précision et la F-mesure sont des mesures largement utilisées dans les évaluations en TALN. La précision est le pourcentage des résultats corrects parmi les résultats obtenus. Le rappel est le pourcentage des résultats corrects parmi les résultats qu'on doit trouver. La F-mesure est la combinaison de la précision et du rappel et leur pondération. La formule de la F-mesure est: [9]

$$\text{F\_mesure} = \frac{2(\text{précision} * \text{rappel})}{\text{précision} + \text{rappel}}$$

Pour le domaine de l'extraction des ENs, les taux de la précision et du rappel sont calculés selon les formules suivantes :

$$\text{Précision} = \frac{Nombre\ d'ENScorrectemnt\ reconnus}{Nombre\ d'ENS\ reconnues}$$

$$\text{Rappel} = \frac{\text{Nombre d'ENS correctement reconnues}}{\text{Nombre d'ENS dans le corpus}}$$

II.10. Les différents types de systèmes

Les systèmes de détection des EN peuvent être classés en trois grands types :

- **Les systèmes fondés sur des règles écrites « à la main »**

Dans ces systèmes, le concepteur doit élaborer un ensemble de règles qui seront ensuite utilisées pour détecter les entités nommées. Historiquement, cette technique fût la première utilisée dans le domaine. Bien que depuis la campagne d'évaluation MUC-6, l'apprentissage automatique ait fait son apparition dans ce domaine, les systèmes à base de règles écrites « à la main » restent encore très utilisés aujourd'hui.

- **Les systèmes à base d'apprentissage automatique**

Ces systèmes utilisent des techniques d'apprentissage automatique pour apprendre un modèle capable d'étiqueter des entités nommées à partir d'un corpus d'apprentissage.

- **Les systèmes mixtes**

Ces systèmes utilisent généralement des lexiques initiaux. Parmi ces systèmes, distingue deux approches. La première consiste à apprendre automatiquement des règles, puis à utiliser un expert pour les réviser. Dans la seconde, un ensemble de règles de base est constitué par le concepteur, puis étendu (semi) automatiquement par inférence, afin d'obtenir une meilleure couverture.

II.11. Quelques problématiques liées aux entités nommées

Comme nous l'avons vu, la notion d'entité nommée est mouvante et fait appel à de nombreux domaines : théorie de la désignation, noms propres, descriptions définies, modèles applicatifs en RI, analyse incrémentale du langage, etc. D'une part, détecter, reconnaître et résoudre les entités nommées, même si leur champ était clairement défini, présente encore des difficultés majeures dans certains cas (entités peu courantes, entités nouvelles dans l'actualité, variantes d'écriture lors de leur traduction, etc.). D'autre part, ces difficultés sont à chaque

fois plus saillantes alors que le périmètre recouvert est continuellement étendu (adresses, produits, événements, etc.). Le besoin de clarifier cette notion et de disposer d'un module de traitement dédié à leur sujet se fait alors de plus en plus ressentir, par exemple pour les applications suivantes :

– Indexation et recherche d'information : les entités nommées détectées dans des documents peuvent permettre de construire des index que pourront exploiter les moteurs de recherche.

– Annotation en rôles sémantiques : dans le cadre d'un mécanisme de compréhension, déterminer les rôles (agent, patient, objet, instrument, lieu, destination, etc.) peut être conditionné par les types d'entités nommées reconnues.

- Question - réponse : le mécanisme par lequel une machine fournit une réponse à une question donnée peut nécessiter de résoudre des entités dans la question, afin de rechercher la réponse dans des bases de connaissances.

– Résolution conjointe d'autres tâches TAL : tokenisation, analyse morpho-syntaxique ou syntaxique, reconnaissance de l'écriture et de la parole, résolution d'anaphores sont des tâches qui peuvent interagir avec la détection ou la reconnaissance des entités nommées.

Ces applications, idéalement, nécessiteraient que toutes les entités nommées soient au préalable automatiquement résolues (donc détectées et reconnues). [10]

II.12. Campagnes d'évaluation des entités nommées

Après avoir présenté les types les plus courants d'entités nommées, la manière dont sont constitués les corpus et les métriques qui permettent d'évaluer les approches pour la reconnaissance d'entités nommées, nous reportons en tableau 2.1, une liste des principales campagnes d'évaluations conduites sur cette problématique, par ordre chronologique. [10]

Date	Compagne	Langue, modalité	Types	Métriques
1996	MUC-6	anglais écrit, rapports	pers, org, loc	f-mesure
1997	MET-1	espagnol,chinois et japonais, écrit journalistique	pers, org, loc, date, heure, montant, pourcent	f-mesure
1998	MET-2	chinois et japonais, écrit journalistique	pers, org, loc, date, heure, montant, pourcent	f-mesure
1999	IREX	japonais,écrit journalistique	pers, org, loc, artefact, date, heure, montant, pourcent	f-mesure
2002	CoNLL-2002	espagnol et flamand, écrit journalistique	pers, org, loc, misc	f-mesure
2003	CoNLL-2003	anglais et allemand, écrit journalistique	pers, org, loc, misc	f-mesure
2006	HAREM	portugais,écrit journalistique	pers, org, loc, temps, oeuvre, événement, abstraction, chose, valeur, autre	pondération d'erreurs
2006	SIGHAN		pers, org, loc, entité géopolitique	f-mesure

2007	ACE07	anglais, arabe et chinois, écrit journalistique et conversationnel	pers, org, loc, bâtiments, entité géo-politique, armes, véhicules	pondération d'erreurs
2007	EVALITA 2007	italien, écrit journalistique	pers, org, loc, entit géopolitique	f-mesure
2008	ACE08	anglais et arabe, écrit journalistique et conversationnel	pers, org, loc, bâtiments, entité géo-politique	pondération d'erreurs
2009	ESTER2	français, oral journalistique	pers, org, loc, temps, montant, fonction, produit	SER
2011	EVALITA 2011	italien, oral journalistique	pers, org, loc, entité géopolitique	f-mesure
2012	ETAPE	français, oral journalistique et conversationnel	pers, org, loc, temps, montant, fonction, produit	SER

Tableau II.3. Caractéristiques des principales campagnes d'évaluation. [10]

II.13. Conclusion

La tâche de reconnaissance des entités nommées a fait cette dernière décennie l'objet d'une attention plus soutenue et suscite aujourd'hui un intérêt certain; elle apparaît en effet comme fondamentale pour diverses applications de TAL participant de l'analyse de contenu, `a l'instar de la recherche et l'extraction d'information, la tâche de question-réponse, le résumé automatique ou encore le fonctionnement des moteurs de recherche, et nombreux sont les travaux se consacrant `à cette tâche, obtenant des résultats plus que probants, et ce pour diverses langues. Aussi, il est d´désormais possible d'affirmer qu'il s'agit d'un des incontournables du traitement automatique des textes.

CHAPITRE III :

Conception et intégration

III-Conception de notre système d'identification automatique d'entités nommées

Chapitre III : Conception de notre système d'identification automatique d'entités nommées

III.1. Introduction

La recherche d'information est un domaine historiquement lié aux sciences de l'information et à la bibliothéconomie qui ont toujours eu le souci d'établir des représentations des documents dans le but d'en récupérer des informations, à travers la construction d'index. L'informatique a permis le développement d'outils pour traiter l'information et établir la représentation des documents au moment de leur indexation, ainsi que pour rechercher l'information.

L'objectif de notre travail est de créer un processus capable de reconnaitre les entités nommées et leurs identifications pour objectif de fournir un accès facile à l'information, cette information étant située dans une masse de documents textuels.

III.2. Processus de recherche d'informations

Dans notre travail, nous utilisons le processus de recherche d'information avec toute les étapes cité dans le chapitre 1, la figure III.1 résume l'étape d'indexation que nous allons suivre.

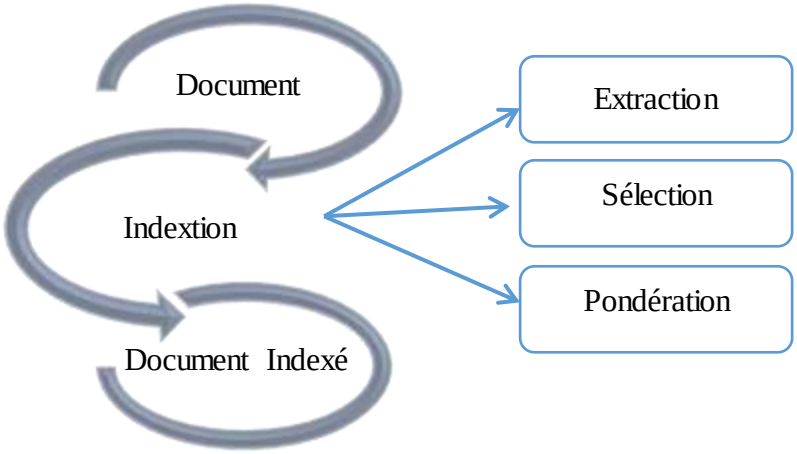


Figure III.1. Indexation d'un document. [22]

III.3. Le moteur de recherche Lucene

C'est un moteur de **recherche** et **d'indexation** développé dans le projet Apache. C'est un logiciel écrit en java et open source signifiant que son code source est libre et accessible gratuitement. Ce logiciel est une librairie de fonctions de recherche dans le contenu textuel des documents. Il inclut une interface de programmation (API).

Lucene est capable de traiter de grands volumes de documents grâce à sa puissance et à sa rapidité dues à l'indexation **[**20**]**. La Figure III.2 décrit le processus d'indexation d'un document avec lucene, il se compose de trois phases :

III-Conception de notre système d'identification automatique d'entités nommées

- La phase d'encapsulation d'un document dont la classe Parsers le transforme sous format d'un objet Document.
- L'analyse s'applique au Document à travers l'analyseur souhaité.
- La création d'index est réalisée par IndexWriter suivant l'emplacement choisi par Directory.

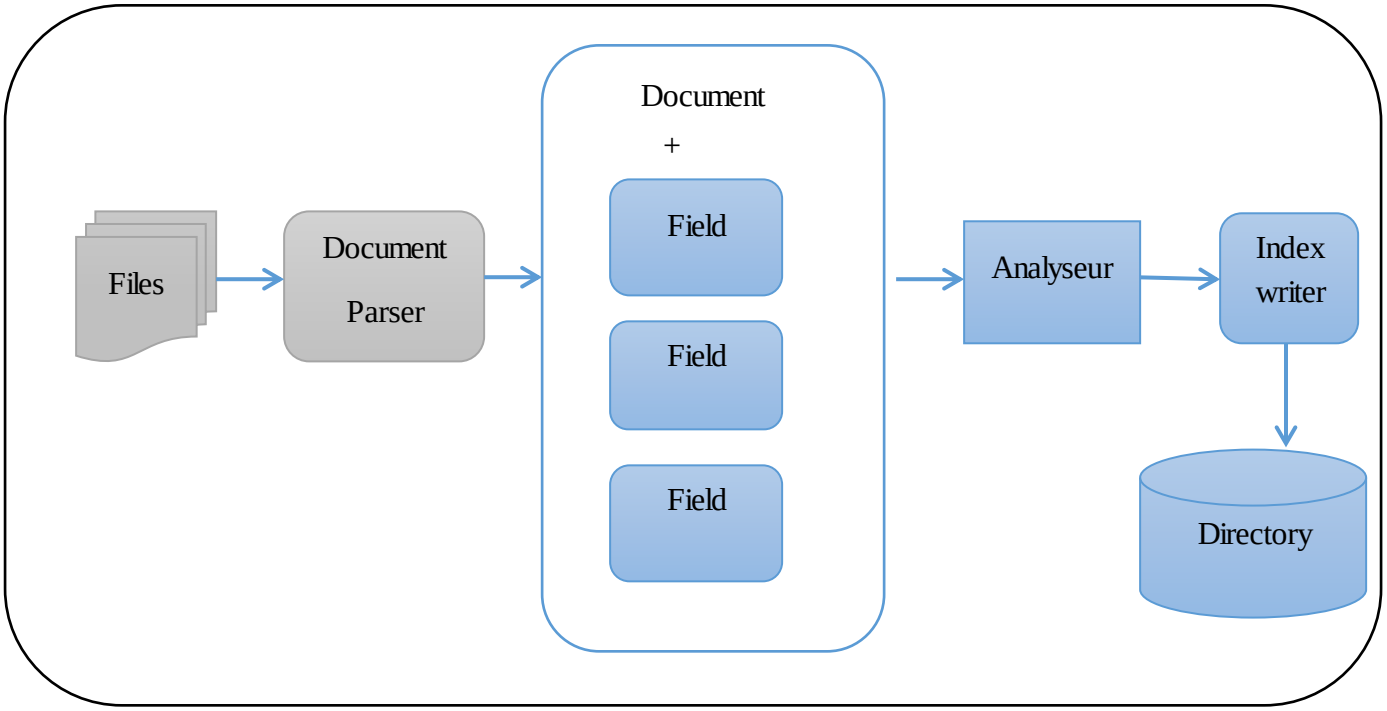


Figure III.2. Processus d'indexation de Lucene. [20]

III.3.1. Architecture et fonctionnement de Lucene

Le moteur de recherche Lucene est caractérisé par deux fonctionnalités ou deux tâches principales qui sont : L'indexation et la recherche.

✓ **Processus d'indexation :**

Permet de Créer un **IndexWriter** utilisé pour écrire le fichier d'index en choisissant un **Analyseur** compatible avec ce dernier.

- **IndexWriter**: cette classe est l'élément central du processus d'indexation. Elle permet de créer un nouvel index (ou ouvrir un index existant), ajouter, supprimer ou mettre à jour les documents dans l'index.
- **Analyzer** : est une classe abstraite livrée avec plusieurs implémentations. Avant l'indexation, le texte est passé à travers l'analyseur spécifié dans le constructeur IndexWriter.

✓ **Processus de recherche :**

Cette phase permet de lire l'index créé à l'aide de **l'IndexReader**, elle permet aussi de créer aussi un **IndexSearcher** prêt à rechercher en choisissant un Analyseur qui va être interrogé par **QueryParser.**

- **IndexSearcher**: est une classe qui ouvre un index en lecture seule, elle nécessite une instance Directory (tenant l'index crée), puis elle offre quelques méthodes de recherche dont certaines sont implémentées dans la classe abstraite Searcher. La plus simple, prend comme paramètres l'objet Query et un nombre entier topN, et retourne un objet TopDocs.
- **TopDocs**: cette classe est un simple conteneur de pointeurs vers les N premiers documents des résultats de recherche, qui correspondent à une requête donnée.

Par ailleurs, par la figure ci-dessous, nous constatons que, Lucene est composé de plusieurs modules afin d'exécuter les processus d'indexation et de recherche :

- **Lucene.analysis**: Classe Analyse est responsable d'analyser un document et Sans analyse, IndexWriter ne peut pas créer d'index
- **Lucene.index**: ce module est principalement responsable de la création de l'index.
- **Lucene.store**: ce module est principalement responsable de la lecture et l'écriture dans l'index.
- **Lucene.QueryParser**: ce module est responsable à l'analyse de la requête.
- **Lucene.search**: ce module est principalement responsable de la recherche dans l'index.
- **Lucene.similarity**: ce module est principalement responsable de la réalisation de la notation de corrélation.

Nous présentons dans la **figure III.3**l'architecture générale de Lucene en détaillant les deux phases principales : l'indexation et la recherche

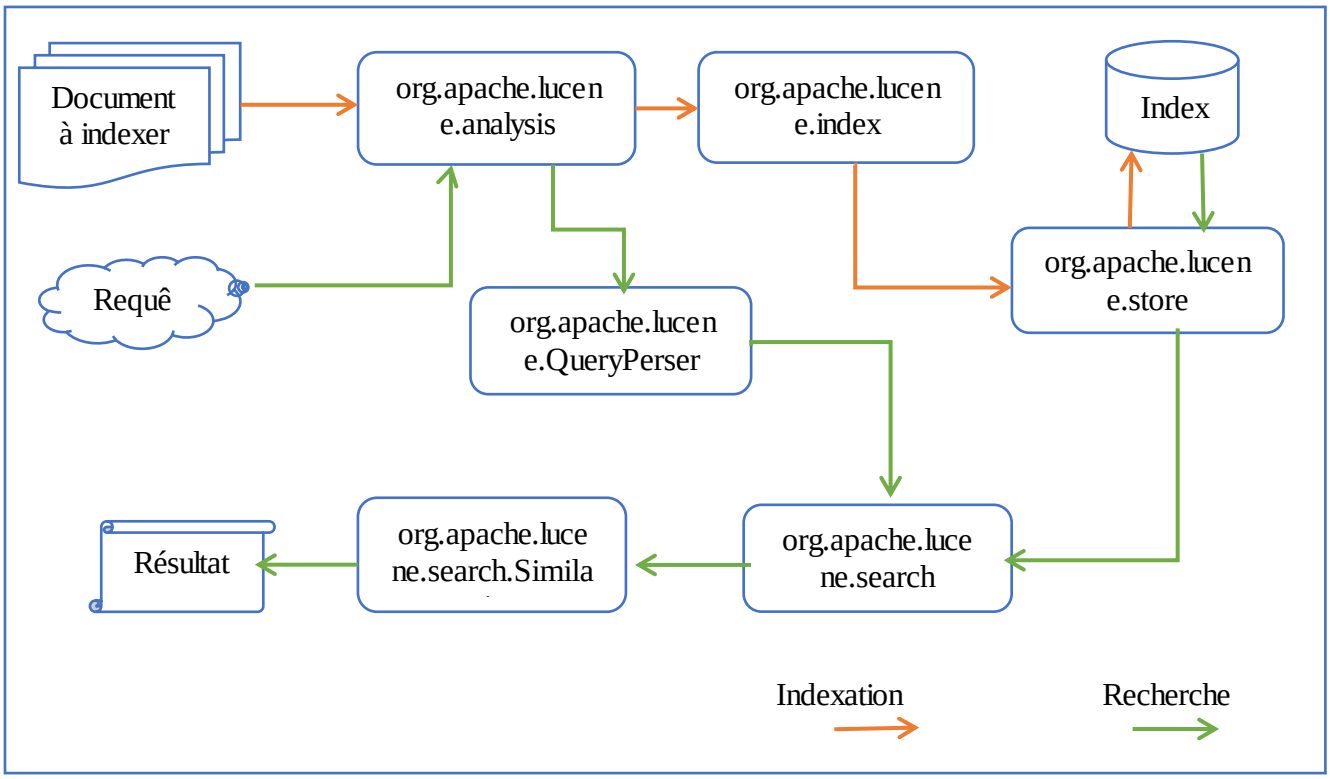


Figure III.3. Architecture générale de Lucene .

III.4.Term Frequency-Inverse Document Frequency Fonction de pondération

III-Conception de notre système d'identification automatique d'entités nommées

La pondération permet d'affecter à chaque terme d'indexation une valeur qui mesure son importance dans le document où il apparaît. Le pouvoir de discrimination des termes pour décrire le contenu des documents n'est pas identique pour tous les termes. Pour trouver les termes du document qui représentent le mieux son contenu, [9] a défini la **fonction de pondération** d'un terme dans un document connue sous la forme de **Tf.Idf**, qui est reprise dans différentes versions par la majorité des SRI. On y distingue :

Un modèle souvent utilise dans des applications de Recherche d'Information (RI) est le modèle Term Frequency-Inverse Document Frequency (TF-IDF). Le poids que ce modèle associe à un certain mot pour un certain document est défini comme le produit de deux valeurs, dont l'une donne au mot considère un certain poids relatif à sa fréquence d'apparition dans le document considéré (c -à-d. poids local) et l'autre donne au mot considère un certain poids relatif au nombre de documents du corpus dans lesquels on retrouve ce mot (c.-à-d. poids global). [5]

III.4.1.Tf (termfrequency) : (pondération locale).

Cette mesure est proportionnelle à la fréquence du terme dans le document. L'idée sous-jacente est que plus un terme est fréquent dans un document, plus il est important dans la description de ce document. Le Tf est souvent exprimé selon l'une des déclinaisons suivantes:

TF = Nombre d'occurrences du terme au sein du document.
Nombre d'occurrence du terme analysé / Nombre de termes total[4]

III.4.2. Idf (Inverse of Document Frequency): (pondération globale)

Mesure l'importance d'un terme dans toute la collection. L'idée sous-jacente est que les termes qui apparaissent dans peu de documents de la collection sont plus représentatifs du contenu de ces documents que ceux qui apparaissent dans tous les documents de la collection. Cette mesure est exprimée selon l'une des déclinaisons suivantes :

IDF *= log (Nombre total de documents / Nombre de documents contenant le terme analysé)* [1]

La mesure « tf * idf» donne une bonne approximation de l'importance du terme dans le document, particulièrement dans les corpus de documents de taille homogène. Cependant, elle ne tient pas compte d'un aspect important du document : sa longueur. En général, les documents les plus longs ont tendance à utiliser les mêmes termes de façon répétée, ou à utiliser plus de termes pour décrire un sujet. Par conséquent, les fréquences des termes dans les documents seront plus élevées, et les similarités à la requête seront également plus grandes. Pour cette raison nous avons utilisé la formule normalisée suivante.

$$tf*idf(t, d) = tf(t, d) * log (N/df(t))$$

Où :

Tf : est la fréquence du terme *'t'* dans le document *'d'*.

[4] http://www.quentinfily.fr/tf-idf-pertinence-lexicale/

III-Conception de notre système d'identification automatique d'entités nommées

Idf : c'est le nombre de document contenant le terme *'t'*.
N: c'est le nombre total de documents dans la collection

Exemple :

Document 1	Document 2	Document 3
Dans le modèle vectoriel les documents et les requêtes sont représentés par des vecteurs d'un espace à n dimensions, ou les coordonnées de chaque vecteur représentent le poids du terme dans la requête ou dans le document, Dans ce modèle, chaque mot à un poids dans chaque document, ce poids représente le degré de pertinence d'un document à une requête se traduit par la fonction de pondération.	Un système de recherche d'information est défini par un langage de représentation des documents qui peut s'appliquer à différents corpus de documents et des requêtes qui expriment un besoin de l'utilisateur et une fonction de mise en correspondance du besoin de l'utilisateur et du corpus de documents pour pouvoir renvoyer aux utilisateurs les documents pertinents pour la requête	Le modèle booléen est basé sur la théorie des ensembles et l'algèbre de Boole. Dans ce modèle, un document est représenté par un ensemble de mots-clés (termes) et la requête de l'utilisateur est représentée par une expression logique composée de termes reliés par des opérateurs logiques : et, ou et le non. Le modèle booléen est le modèle le plus ancien et également le plus simple en RI.

L'exemple porte sur le document 1 (soit ***d1***) et le terme analysé est « modèle » (soit ***t1*** = modèle).

Calcul de Tf

- Le document 1 (**d1)** contient 69 mots (termes).
- Le document 2 (**d2)** contient 62 mots (termes).
- Le document 3 **(d3)** contient 70 mots (termes).

$$\mathbf{Tf_i} = \frac{\sum ti}{\sum jni,j}$$

d j un document,

ti un terme,

$\mathbf{n_{i,j}}$ le nombre d'occurrences du terme ti dans dj.

$$Tf_{1,1} = \frac{\sum t1,1}{\sum jnj,1} = \frac{2}{69} \approx 0{,}029$$

$$Tf_{1,2} = \frac{\sum t1,2}{\sum jnj,2} = \frac{0}{62} = 0$$

$$Tf_{1,3} = \frac{\sum t1,3}{\sum j\ nj,3} = \frac{4}{70} \approx 0{,}057$$

Calcul de Idf

N est le nombre total de documents dans le corpus

| {dj : ti ∈ dj}| le nombre de documents contenant le terme **ti**

$$\mathbf{Idf_i = \log \frac{N}{|\{dj : ti \in dj\}|}}$$

$$Idf_1 = \log \frac{|D|}{|\{dj : t1 \in dj\}|} = \log \frac{3}{2} \approx 0{,}176$$

Calcul de tf-idf : multiplication de deux mesures :

$$\mathbf{Tfidf_{i,j} = tfi_{i,j} \cdot idf_i}$$

On obtient :

$$tfidf_{1,1} = \frac{2}{69} \cdot \log \frac{3}{2} \approx 0{,}005$$

$$tfidf_{1,2} = 0 \cdot \log \frac{3}{2} = 0$$

$$tfidf_{1,3} = \frac{4}{70} \cdot \log \frac{3}{2} \approx 0{,}010$$

Donc le troisième document apparaît ainsi comme « le plus pertinent »

III.5. Indexation des données

Un **index** est une liste alphabétiquement ordonnée, En informatique, un index est une liste ordonnée qui permet un accès plus rapide à une ligne spécifique d'une table d'une base de données à partir de la valeur d'une ou de plusieurs colonnes de cette ligne.[5]

Indexation Lucene a recours au package nommé org.apache.lucene.index contenant les classes IndexWriter et IndexReader.

L'index est une structure de données stockée sur le système de fichiers. Les documents de l'utilisateur vont être ajoutés à l'index, contient une série de documents, les termes retenus après l'analyse, les champs et les segments. Tout document de Lucene est composé de champs divers : titre, auteur, contenu (contents). Chaque champ contient un nom et une valeur. A l'intérieur du champ, on retrouve une séquence de termes. [23]

Exemple : titre : langage de programmation
Nom du champ : titre valeur du champ : langage de programmation

III.5.1. Indexation avec lucene

Lucene est capable de lire un très grand nombre de formats : PDF, Word, HTML, XML et TXT. Au moment de l'indexation, il ne traitera uniquement que le contenu textuel des documents. Voici un schéma qui montre comment sont exploités les documents au moment de l'indexation. Avant d'être indexée, la structure syntaxique et le texte des documents sont analysés .De plus, lors de l'indexation, il va assigner à chaque document de l'index un identifiant unique (Document ID). Après la création d'un index, il est possible de rajouter ou supprimer des documents avec l'instance IndexWriter. Les données de l'index sont lues par le biais de la classe IndexReader. Il est stocké dans un répertoire unique. Son emplacement,

[5] http://www.larousse.fr/dictionnaires/francais/index/42560

déterminé par la classe Directory provenant du package org.apache.lucene.store, est situé dans le système de fichiers. L'utilisateur aura recours à l'implémentation : FSDirectory comme pour la base de documents choisie pour la mise en œuvre l'index est composé de segments, pouvant être considérés comme des sous-index bien qu'ils ne soient pas entièrement indépendants. Lucene va assigner à chaque document de l'index un identifiant unique (Document ID). [23]

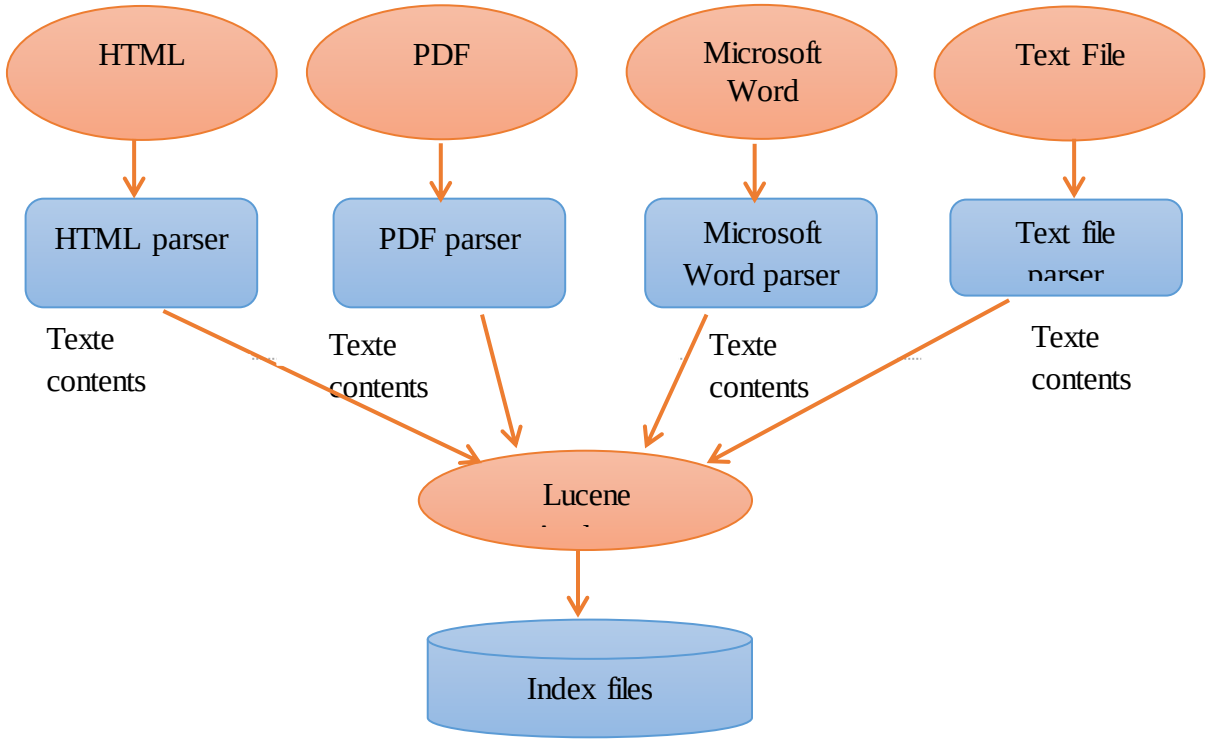


Figure III.4. Architecture d'indexation [23]

III.5.2. Indexation de textes

Pour un texte, un index très simple à établir automatiquement est la liste ordonnée de tous les mots apparaissant dans les documents avec la localisation exacte de chacune de leurs occurrences ; mais un tel index est volumineux et surtout peu exploitable.

L'indexation automatique tend donc plutôt à rechercher les mots qui correspondent au mieux au contenu informationnel d'un document. On admet généralement qu'un mot qui apparaît souvent dans un texte représente un concept important. Ainsi, la première approche consiste à déterminer les mots représentatifs par leur fréquence. Cependant, on s'aperçoit que les mots les plus fréquents sont des mots fonctionnels (mots vides). En français, les mots « de », « un », « les », etc. sont les plus fréquents. En anglais, ce sont « of », « the », etc.

```
Début
Char [] cArray=null;
        String docpath;
        String ligne  = null;
        String ligne  = null;
        InputStream flux=new FileInputStream (docpath);
```

III-Conception de notre système d'identification automatique d'entités nommées

```
InputStreamReader lecture=new InputStreamReader (flux);
        BufferedReader buff=new BufferedReader (lecture);
        LineNumberReader l = new LineNumberReader (buff);
        Créer le fichier d'index ;
Tant que (ligne=l.readLine ())!=null) faire
        Eliminer les mots vides ;
        Eliminer les caractères spéciaux ;
        Rendre toute les mots en minuscule ;
        Création de vecteur mot ;
Ftq ;
Fin
```

Algorithme III.1. Vectorisation des fichiers

Une autre opération est ensuite couramment appliquée lors de l'indexation. Elle consiste à effacer les terminaisons (flexions de nombre, genre, conjugaison, déclinaison) afin de retrouver les racines des mots. Cette opération est appelée lemmatisation (stemming). Ce procédé permet de relever les fréquences en cumulant les nombres d'occurrence des variations des mêmes mots. Il existe plusieurs algorithmes de ce traitement, Par exemple l'algorithme de Porter[6] (pour l'anglais) et L'algorithme Carry[7] (pour le français).

[6]L'algorithme développé par Porter se compose d'une cinquantaine de règles de racinisation classées en sept phases successives, Les mots à analyser passent par tous les stades et, dans le cas où plusieurs règles pourraient leur être appliquées

[7]L'algorithme de *Carry* se déroule en diverse étapes par lesquelles les mots à traiter passent successivement. Selon les règles, quand l'analyseur reconnaît un suffixe de la liste, soit il le supprime, soit il le transforme. C'est ici aussi le suffixe le plus long qui détermine la règle à appliquer

III-Conception de notre système d'identification automatique d'entités nommées

```
Pour chaque terme
Obtention d'une forme tronquée du mot
    Suppression des flexions
    Suppression des suffixes
Obtention de la forme canonique
    Pour un verbe: sa forme à l'infinitif
    Pour un nom, adjectif, article, ... : sa forme au masculin singulier
Fin pour chaque
```

Algorithme III.2. Racinisation et Lemmatisation

Les moteurs de recherche de première génération s'appuient sur des formules de pondération, généralement pour affecter un poids élevé aux termes non-distribués uniformément au sein du corpus. Il existe un grand nombre de formules de pondération dont le but est de distribuer le poids pour contribuer à la différenciation informationnelle des documents les formules de pondération les plus connues sont TF-IDF (term frequency. inverse document frequency) que nous utilisons dans notre système. L'algorithme suivant présente le calcul du TF/ IDF tél que nous l'avons implémenté :

```
Pseudo code 1:
Pour (i = 0 i <nombre de mots uniquei ++)
  Pour (j = 0 j <numéro de Documents j ++)
Tfidf = fij _ log (nombre de Documents = ni)
Pour (s = 0 s <nombre de mots unique s ++)
  FijTemps = nomber_de_occurrences_demots_S_dans _document_J
  TfidfTemps = Tempsfixe * log (nombre de Documents / dfi)
  SummTfidf + = (tfidfTemps) 2
fin
   A [i, j] = tfidf / summTfidf
 fin
fin
```

Algorithme III.3. TF-IDF (term frequency - inverse document frequency)

III.6. L'annotation

III.6.1. L'annotation d'entités nommées

L'annotation d'entités nommées est une activité d'extraction d'information dans des corpus documentaires. Elle consiste à rechercher des objets textuels catégorisables dans des classes telles que noms de personnes, noms d'organisations ou d'entreprises, noms de lieux, quantités, distances, valeurs, dates, etc.

L'annotation est le processus qui consiste à relier des informations complémentaires au contenu textuel d'un document. Nous avons parlé dans le chapitre précédent en détaille sur l'annotation des entités nommées.

Ce type d'annotation a été réalisé sur une partie du corpus Reuters 21578[8], L'annotation a été réalisée sur Trente-et-un fichiers, soit un total de 26,7 Mo. Avec un logiciel Stanford nommée entité Recognizer (NER)[9].

Exemple :

« Henri a acheté 300 actions de la société AMD en 2006 »

Henri → Personne 2006 → Date

AMD → Organisation

III.6.2. Stanford Nommée Entité Recognizer (NER)

Stanford NER est une implémentation Java d'un Reconnaissance d'Entité Nommée. La reconnaissance de l'entité nommée (NER) étiquette les séquences de mots dans un texte qui sont le nom des choses, tels que les noms de personnes, d'organisation et de localisation. Il est livré avec des extracteurs de fonctions bien conçus pour la reconnaissance d'entité nommée, et beaucoup d'options pour définir des extracteurs de caractéristiques. Dans le cadre du téléchargement, les reconnaisseurs d'entités sont bien reconnus pour l'anglais, en particulier pour les 3 classes (PERSONNEL, ORGANISATION, L'EMPLACEMNT).

Stanford NER est également connu sous le nom de CRF Classifier. Le logiciel fournit une implémentation générale de modèles de séquence de champ aléatoire conditionnel (CRF) de chaîne linéaire (ordre arbitraire). C'est-à-dire en formant vos propres modèles sur les données marquées, vous pouvez effectivement utiliser ce code pour construire des modèles de séquence pour NER ou une autre tâche.

[8] http://igm.univ-mlv.fr/~mconstan/enseignement/m2pro/tal/tal-td2/node1.html
[9] https://nlp.stanford.edu/software/CRF-NER.shtml

III-Conception de notre système d'identification automatique d'entités nommées

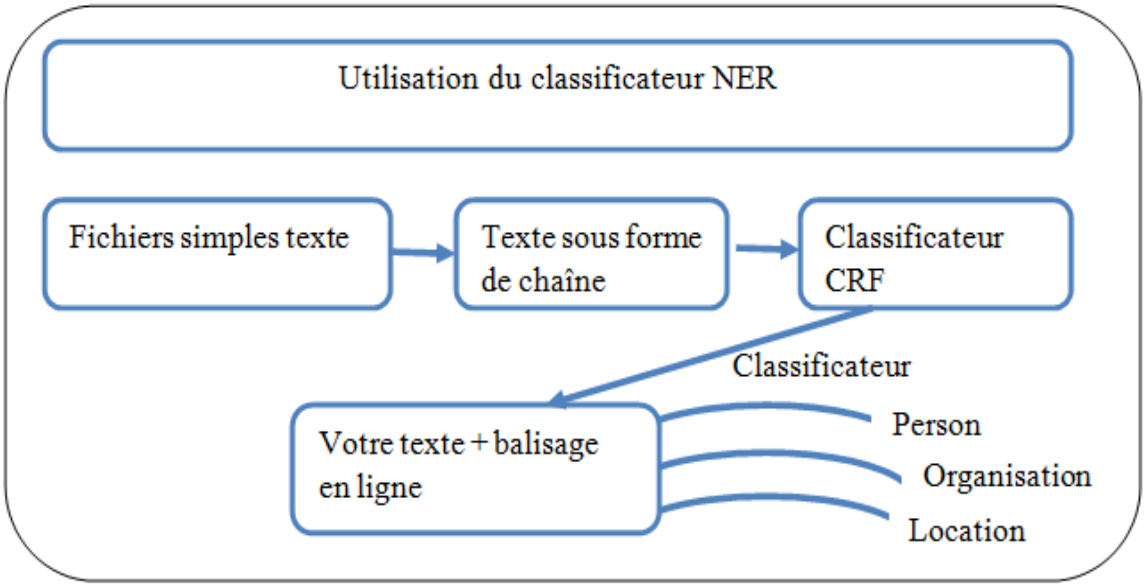


Figure III.5. Fonctionnent de NER

Exemple: les résultats de NER sont comme suit :

```
Entités :
        {
                PERSON:''
                LOCATION:''
                ORGANIZATION:''
        }
```

III.6.3. Utilisation programmée via l'API

Stanford NER peut être appelé à partir de votre propre code. Le fichier NERDemo.java[10] inclus dans la distribution illustre plusieurs façons d'appeler le système par programme.

III.6.4. Utilisation programmée via un service

Stanford NER peut également être configuré pour fonctionner en tant que serveur qui écoute sur un socket.

III.6.5. Les modèles

Compris avec Stanford NER sont un modèle de 4 classes formé sur CoNLL 2003, un modèle de classe 7 formé sur les ensembles de données de formation MUC 6 et MUC 7, et un modèle de 3 classes formé sur les deux ensembles de données et des données supplémentaires (y compris ACE 2002 et quantités limitées de données internes) sur l'intersection de ces ensembles de classes.

3 Classes : Lieu, Personne, Organisation.

4 Classes: Emplacement, Personne, Organisation, Divers.

7 Classes: Lieu, Personne, Organisation, Argent, Pourcentage, Date, Heure.

[10] https://nlp.stanford.edu/software/CRF-NER.shtml#Download

Ces modèles utilisent chacun des caractéristiques de similarité de distribution, qui offrent un gain de performance au prix d'augmentation de leur taille et de leur durée d'exécution. Aussi disponibles sont les mêmes modèles qui manquent ces fonctionnalités.

III.7. La Recherche

Recherche Le modèle standard espace vectoriel (the vector space model) est utilisé par le moteur de recherche Lucene. Il a pour but de donner plus d'importance aux termes apparaissant souvent (term frequency) dans le document, mais qui sont relativement rares dans l'ensemble de la base de documents. Les documents et requêtes sont représentés comme des vecteurs. Si un terme apparaît dans un document, sa valeur dans le vecteur est non-nulle. Le vecteur se présente sous cette formule : V = [w1, w2,…., wn] où w est le poids de chaque terme.

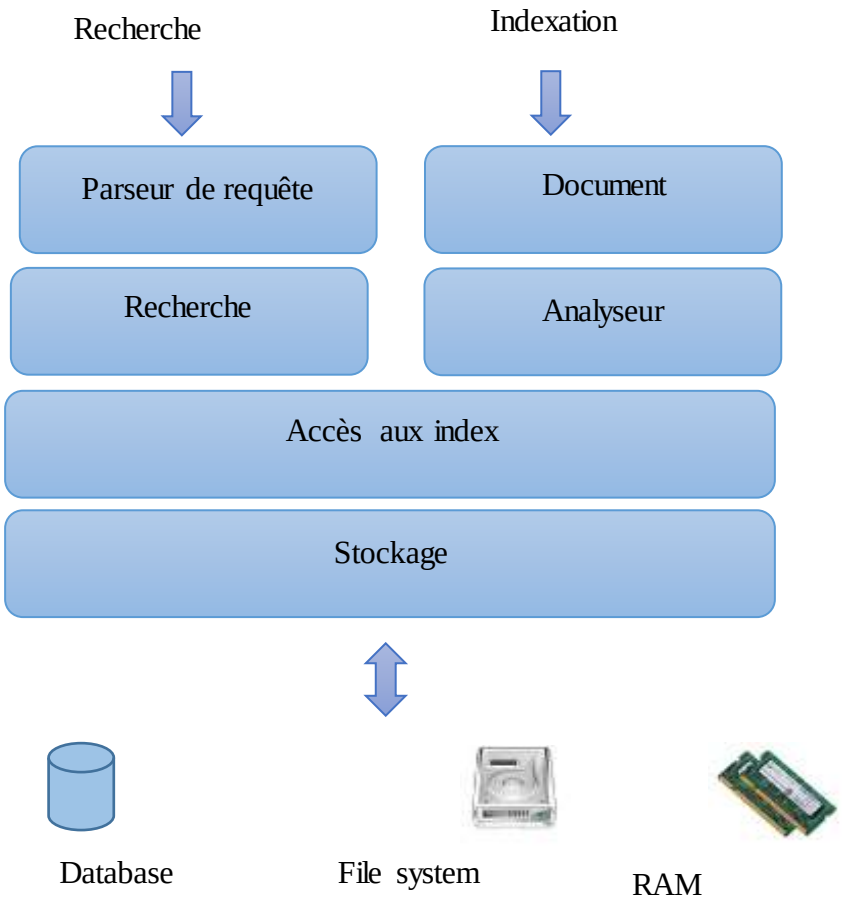


Figure III.6. Architecture et organisation de Lucene [23]

III-Conception de notre système d'identification automatique d'entités nommées

L'algorithme suivant présente l'étape de recherche.

```
Pour chaque terme
Créer une requête
Questionnez l'index
Pour chaque résultat obtenu (pour chaque document)
Obtenir le terme vector de fréquence
Remplir deux structures de données:
docVectors: document - vector
docSumOfFreq: document - somme du terme fréquences
Fin pour chaque
Fin pour chaque
```

Algorithme III.4. La recherche

Cependant, on peut constater que le modèle booléen est inclus dans Lucene dans le sens où le document correspond ou ne correspond pas à la requête demandée. C'est lui qui attribue la pertinence des documents. Si la requête est bonne, il retourne les scores et un ensemble de documents sinon il retourne « false » c'est-à-dire qu'aucun résultat ne s'affiche. [23]

Une maniéré particulière d'éffèctuer la recherche en exploitant l'annotation est l'annotation recherche par types annotés (Typed Annotated Search, TAS) présente dans déférents travaux. Ces approches de recherche proposent de guider la recherche d'une information en utilisant son type, par exemple, le nom d'une personne. Les techniques utilisées reposent sur les outils d'extraction d'information déjà disponible pour extraire les types des données. Un exemple est présenté dans la Figure III.7 : « Trouver l'inventeur de la télévision », la recherche se fera en cherchant les noms de personnes en se basant sur le mot clé « invente » et « télévision ».

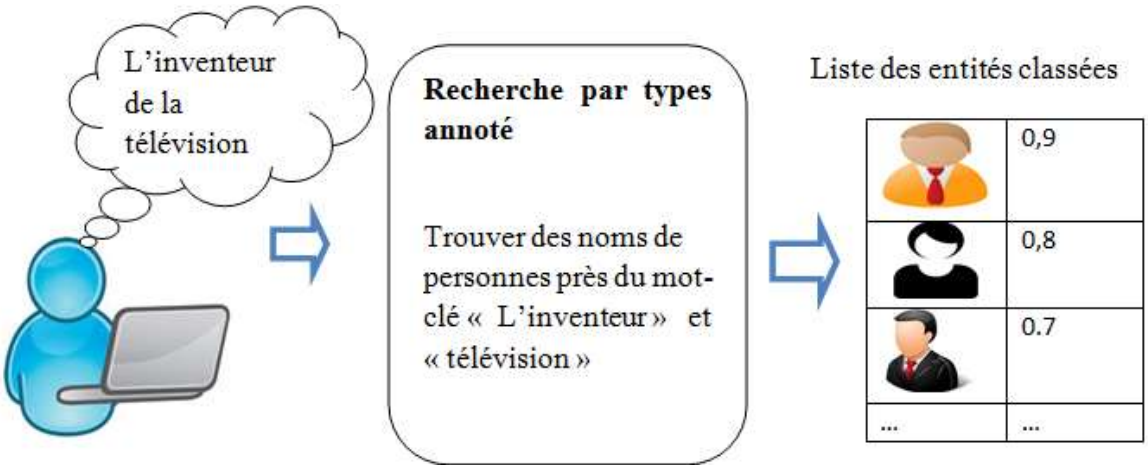


Figure III.7. Recherche par type annoté

III-Conception de notre système d'identification automatique d'entités nommées

III.8. Classes

III.8.1. Classes d'indexation

- **IndexWriter**

La classe IndexWriter est le composant central du processus d'indexation. Cette classe crée un nouvel index et ajoute des documents à un index existant. On peut se la représenter comme un objet par lequel on peut écrire dans l'index mais qui ne permet pas de le lire ou de le rechercher.

- **Directory**

La classe Directory représente l'emplacement de l'index de Lucene. IndexWriter utilise une des implémentations de Directory.
FSDirectory : est une implémentation de Directroy qui stocke les index dans le système de fichiers, ce qui est utile pour les gros index. [11]

RAMDirectory : prend toutes ses données en mémoire. Cela peut être utile pour de plus petits indices qui peuvent être pleinement chargés en mémoire et peuvent être détruits sur la fin d'une application. [23]

- **Analyzer**

Avant que le texte soit dans l'index, il passe par l'Analyser. Celui-ci est une classe abstraite qui est utilisée pour extraire les mots importants pour l'index et supprime le reste. Cette classe tient une part importante dans Lucene et peut être utilisée pour faire bien plus qu'un simple filtre d'entrée. [23]

Analyseur	Opérations effectuées sur les données textuelles
WhitespeceAnalyzer	Découpe en signe sur la base de l'espace
SimpleAnalyzer	Découpe sur la base des caractères autre que des lettres et met le texte en minuscule
StopAnaluzer	Supprime les mots d'arrêt (inutiles pour la recherche) et met le texte en minuscules
StandartAnalyzer	Découper le texte sur la base d'une grammaire sophistique qui reconnait : les adresses électroniques les acronymes, les caractères chinois, coréens, japonais, les combinassions alphanumériques que (liste de non- exclusive) met le texte en minuscule et supprimé le mot d'arrêt.

Tableau III.1. Les Analyseur fournis par Lucene. [3]

[11] https://www.ibm.com/developerworks/java/library/os-apache-lucenesearch/index.html, Using Apache Lucene to search text

- **Document**

La classe Document représente un rassemblement de champs. Les champs d'un document représentent le document ou les métadonnées associées avec ce document. La source originelle (comme des enregistrements d'une base de données, un chapitre d'un livre, etc.) est hors de propos pour Lucene. Les métadonnées comme l'auteur, le titre, le sujet, la date, etc. sont indexées et stockées séparément comme des champs d'un document.

- **Field**

Chaque document est un index contenant un ou plusieurs champs, inséré dans une classe intitulé Field. Chaque champ (field) correspond à une portion de donnée qui est interrogé ou récupéré depuis l'index durant la recherche.

Option	Descripteur
Field.Store.yes	Utilisé pour stocké la valeur des champs. Adapté pour les champs affichés dans les résultats de recherche, chemin du fichier et URL, par exemple
Field.Store.No	La valeur du champ n'est pas stockée – le corpus d'un courrier électronique, par exemple
Field.index.No	Adapté pour les champs qui ne sont pas requêtes – souvent utilisé avec les champs stockés, comme le chemin du fichier
Field.index.ANALYZED	Utilisé pour les champs indexés et analysés – le corpus et le sujet d'un courrier électronique, par exemple
Field.index.No_ANALYZED	Utilisé pour les champs indexés mais non analysés. Il préserve complètement la valeur d'origine du champ - comme par exemple les informations personnelles (dates, noms)

Tableau III.2. Détail des métadonnées de Field. [3]

III.8.2. Classes de recherche

- **IndexSearcher**

La classe IndexSearcher est à la recherche ce qu'IndexWriter est à l'indexation. On peut se la représenter comme une classe qui ouvre un index en mode lecture seule.

- **Term**

Un terme est une unité basique pour la recherche, similaire à l'objet field. Il est une chaîne de caractère : le nom du champ et sa valeur. Notez que les termes employés sont aussi inclus dans le processus d'indexation.

- **Query**

La classe Query est une classe abstraite qui comprend BooleanQuery.

- **TermQuery**

C'est la méthode la plus basique d'interrogation de Lucene. Elle est utilisée pour égaliser les documents qui contiennent des champs avec des valeurs spécifiques.

- **QueryParser**

La classe QueryParser est utilisée pour générer un décomposeur analytique qui peut chercher à travers un index.

- **Hits**

La classe Hits est un simple conteneur d'index pour classer les résultats de recherche de documents qui apparaissent pour une interrogation donnée. Pour des raisons de performances, les exemples de classement ne chargent pas depuis l'index tous les documents pour une requête donnée, mais seulement une partie d'entre eux.

III.9. Conclusion

Dans ce chapitre nous avons présenté notre méthode de conception ainsi que l'architecture générale de notre système. Aussi nous avons présenté le système Lucene qui représente une libraire de recherche open source très populaire provenant d'Apache qui fournit des fonctions d'indexation et de recherche très puissante nécessaire au bon fonctionnement de notre système.

Nous abordons dans le chapitre suivant l'étape d'implémentation de notre logiciel.

CHAPITRE IV :

Implémentation et mise œuvre

Chapitre IV : Implémentation et mise œuvre

IV.1. Introduction :

Dans ce chapitre, nous allons présenter l'objectif de notre travail qui repose sur la proposition d'un programme d'identification d'entité nommées. Nous commençons tout d'abord par la présentation de l'environnement de développement, en détaillant les différents outils utilisés ensuite nous présenterons les différentes interfaces de notre programme.

IV.2. Environnement de travail :

L'implémentation et les tests de notre application ont été réalisés dans l'environnement matériel et logiciel suivant :

IV.2.1. Ressources utilisées :

Les ressources physiques exploitées :

- ✓ Processeur Intel(R) Pentium® CPU B960 @ 2.20GHZ.
- ✓ Mémoire vive d'une capacité de 4.00 Go.

Et comme ressource logicielle, nous avons utilisé :

- ✓ Système d'exploitation : Windows7.
- ✓ Langage de programmation : JAVA.
- ✓ L'EDI : NetBeans de version 8.1

Notre choix s'est porté sur cet EDI car il permet d'intégrer une interface graphique, en utilisant la syntaxe du langage JAVA. Il offre au programmeur un environnement intégré pour la programmation orientée objet, visuelle et événementielle.

IV.2.2. Pourquoi JAVA :

Java est un langage de programmation et une plate-forme informatique créée par Sun Microsystems en 1995, racheté plus tard par Oracle. Il s'agit de la technologie sous-jacente qui permet l'exécution des applications modernes sur différentes plateformes. La portabilité, des programmes Java sur différents systèmes d'exploitation, représente son atout principal. Java est utilisée sur plus de 850 millions d'ordinateurs de bureau et un milliard de périphériques dans le monde, dont des périphériques mobiles et des systèmes de diffusion télévisuelle.

Java est un langage de programmation très utilisé, notamment par un grand nombre de développeurs professionnels, ce qui en fait un langage incontournable actuellement.

On a travaillé avec java car il a beaucoup de caractéristiques parmi lesquelles :

- o Son excellente portabilité: une fois votre programme crée, il fonctionnera automatiquement sous Windows, Mac, Linux etc.
- o on peut faire de nombreux types de programmes avec Java:
  - des applications sous forme de fenêtre ou de console ;
  - des applets, qui sont des programmes Java incorporé à des pages Web ;

- des applications pour appareils mobiles, comme les Smartphones, avec Java ME (Java Micro Edition) ;
- des sites Web dynamiques avec J2EE (Java 2 Entreprise Edition) ;
- et bien d'autre : JMF (Java Media Framework), J3D pour la 3D...

IV.2.3. Pourquoi NetBeans ?

NetBeans est un environnement de développement intégré (EDI), open source et multi-langue, créé par Sun et racheté par Oracle. Il a la particularité d'être multi plateforme: il est compatible avec Windows, MacOs, Linux et Solaris. Il permet d'intégrer une interface graphique en utilisant la syntaxe du langage JAVA.

De licence Open Source, NetBeans permet de développer et déployer rapidement et gratuitement des applications graphiques Swing, des Applets, des JSP/Servlets, de l'architecture J2EE, dans un environnement fortement personnalisable.

IV.3. Architecture de L'application

Nous présentons dans la (Figure **IV.1**) l'architecture globale de notre application :

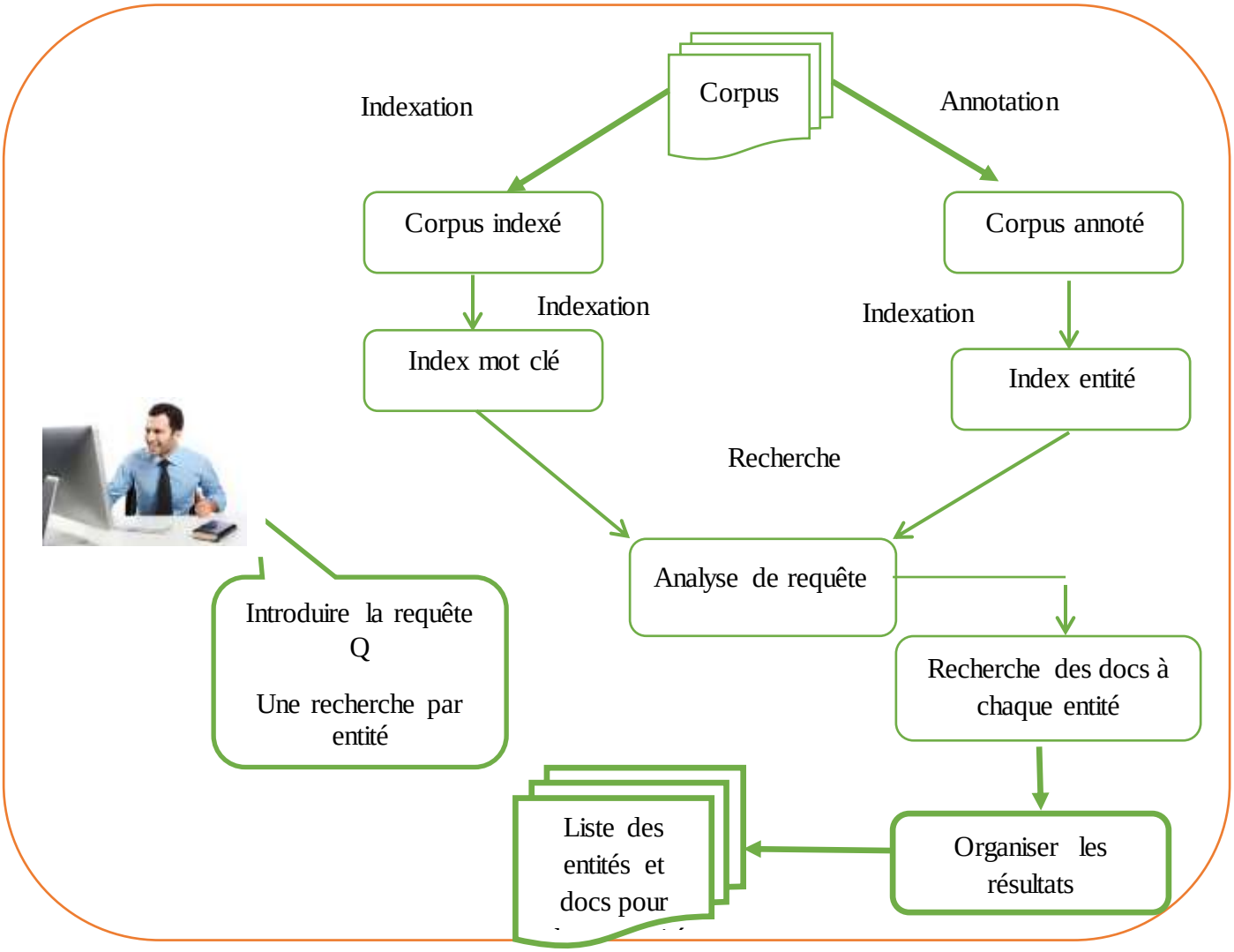


Figure IV.1. Architecture de L'application

Nous considérons, un corpus de documents semi ou non structure, Notre approche consiste à faire un prétraitement pour préparer les informations au traitement en ligne selon la Figure **IV.1** précédente :

Nous commençons par indexer notre corpus, on utilisons le moteur de recherche open source Lucene. Puis on annote le corpus en utilisant un système d'annotation qui est dans notre cas Stanford Nommée Entité Recognizer (NER), pour extraire les entités, leurs types.

Nous créons différents index pour stocker les informations tél que l'index des mots clés et l'index des entités,

Nous commençons par analyser la requête Q pour connaitre son type. L'idée est de faire une diversification des interprétations de la requête pour retourner un ensemble d'entités pertinentes et diverses

IV.4. Corpus de teste

Ce mémoire tente d'évaluer notre proposition d'intégration un processus capable d'identifier des entités nommées, Pour ce faire, on a utilisé un corpus qui il est Reuters 21578.

La collection Reuters 21578 contient 31 documents, et de tailles totales de 26,7 Mo, de l'agence de presse Reuters, c'est est un ensemble de 21 578 nouvelles publiées dans le journal Reuters en 1987, qui sont classées selon 135 catégories thématiques, principalement dans le domaine des affaires et de l'économie.

Il existe 5 groupes de catégories qui étiquettent les documents Reuters-21578: ÉCHANGES, ORGUES, PERSONNES, LIEUX ET SUJET.

```
<!-- 23-Jan-97 This is v1.1 of lewis.dtd, acorrected version by
Chris Brew  -->
<!ELEMENT LEWIS O O (REUTERS+)>
<!ELEMENT REUTERS - -
(DATE,(UNKNOWN|MKNOTE)*,TOPICS,PLACES,PEOPLE,ORGS,EXCHANGES,COM
PANIES,UNKNOWN?,TEXT)>
<!ELEMENT HEAD - - (#PCDATA)>
<!ELEMENT DATE - - (#PCDATA)>
<!ELEMENT TOPICS - - (D|#PCDATA)*>
<!ELEMENT PLACES - - (D|#PCDATA)*>
<!ELEMENT PEOPLE - - (D|#PCDATA)*>
<!ELEMENT ORGS - - (D|#PCDATA)*>
<!ELEMENT EXCHANGES - - (D|#PCDATA)*>
<!ELEMENT COMPANIES - - (D|#PCDATA)*>
<!ELEMENT TEXT - - (TITLE|BODY|AUTHOR|DATELINE|#PCDATA)+>
<!ELEMENT TITLE - - ANY>
<!ELEMENT BODY - - ANY>
<!ELEMENT DATELINE - - (#PCDATA)>
<!ELEMENT AUTHOR - - (#PCDATA)>
<!ELEMENT D - - (#PCDATA)>
<!ELEMENT UNKNOWN - - (#PCDATA)>
<!ELEMENT MKNOTE - - (#PCDATA)>
<!ATTLIST REUTERS
              OLDID CDATA #REOUIRED
```

```
<!ATTLIST TOPICS
          TYPE CDATA "DSET">
<!ATTLIST PLACES
          TYPE CDATA "DSET">
<!ATTLIST PEOPLE
          TYPE CDATA "DSET">
<!ATTLIST EXCHANGES
          TYPE CDATA "DSET">
<!ATTLIST ORGS
          TYPE CDATA "DSET">
<!ATTLIST COMPANIES
          TYPE CDATA "DSET">

<!ATTLIST TEXT
          TYPE (NORM|BLAH|UNKNOWN|UNPROC|BRIEF) NORM>
<!ATTLIST TEXT
          TYPE (NORM|BLAH|UNKNOWN|UNPROC|BRIEF) NORM>
<!ENTITY lt CDATA "&#38;lt;">
<!ENTITY amp CDATA "&amp;">
<!-- The followingdefinitions are for
downlinelarkuptokenisation -->
<!ELEMENT W - - (#PCDATA)>
<!ATTLIST W
```

Figure IV.2. Exemple du corpus Reuters

IV.5. Présentation de l'application :

Dans cette partie on va décrire les différentes parties de notre application coté interface graphique et les différentes opérations de chaque bouton et menu.

La figure suivante représente l'interface principale de notre application :

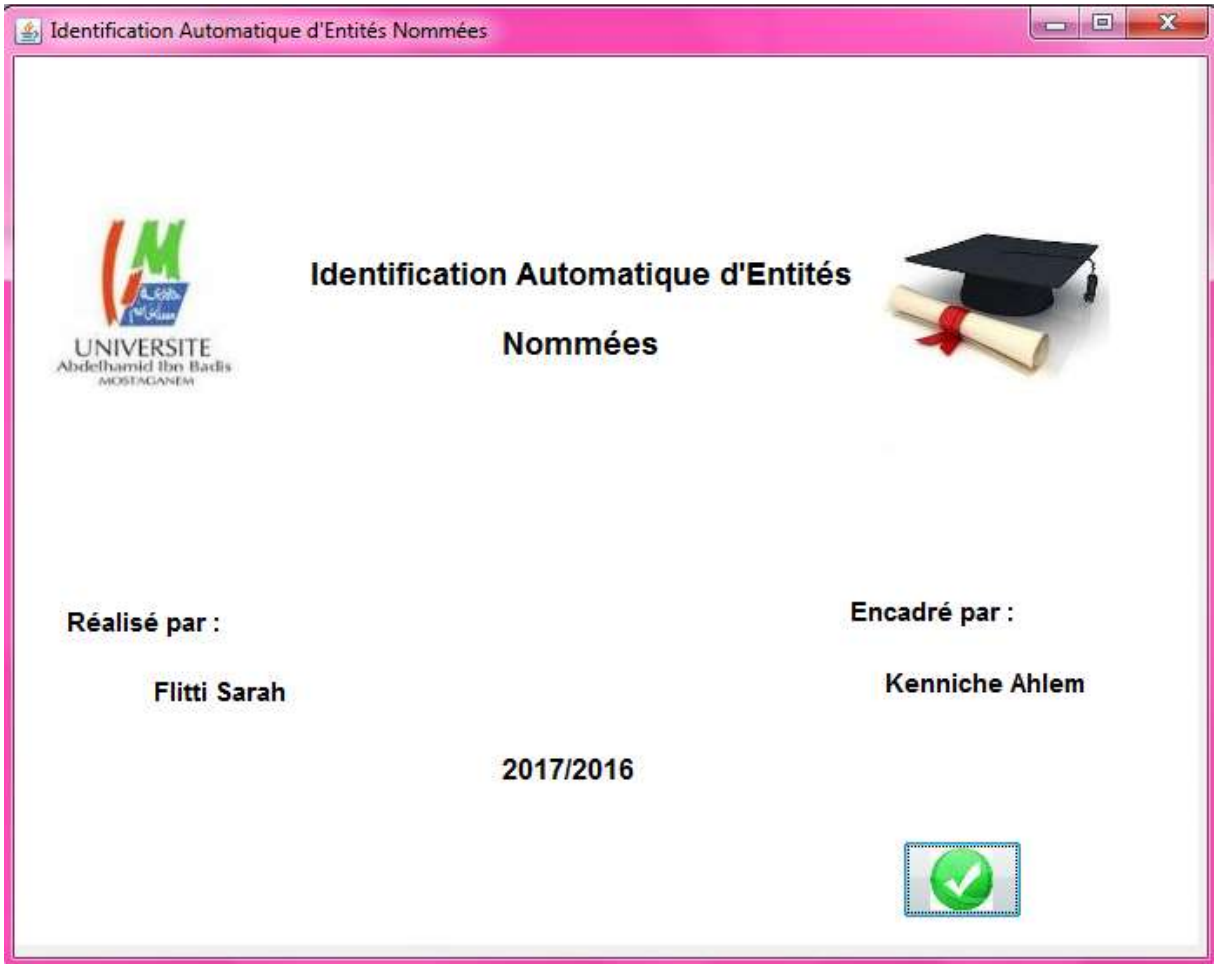


Figure IV.3. L'interface principale de notre application

IV.6. Menu principale

Notre appliquassions est composé de barre d'outille suivant :

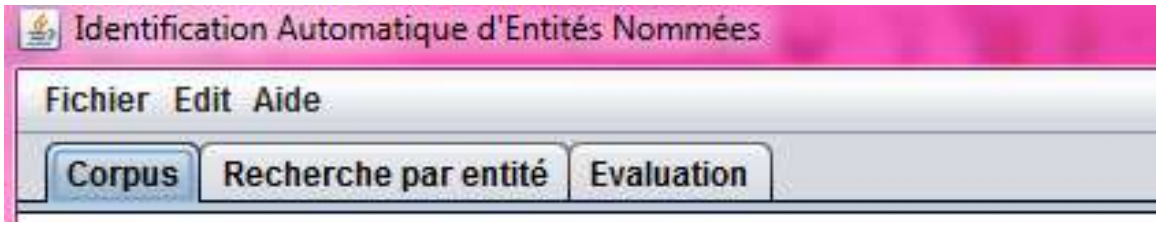


Figure IV.4. Barre d'outille de notre application

IV.6.1. Corpus

L'interface dédiée pour les différentes tâches générales sur un corpus telles que l'ouverture du corpus.

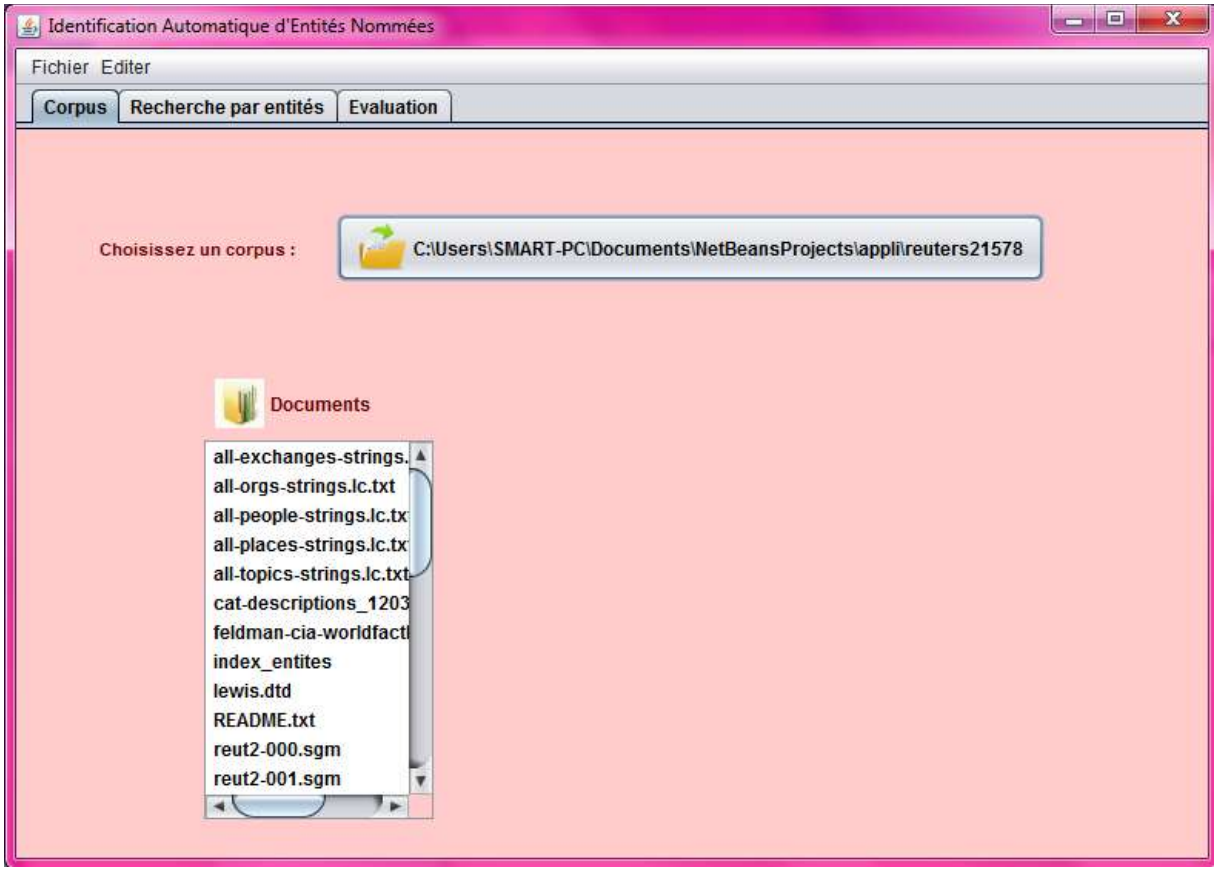


Figure IV.5. Interface de consultation du corpus

IV.6.2. Recherche par entité

IV.6.2.1. Indexation

Pour lancer l'indexation du corpus sélectionné, il faut clique sur le Botton Indexer. Nous présentons dans la figure suivante l'interface dédiée à l'indexation de notre corpus par l'analyseur lucene «**StandardAnalyzer**» :

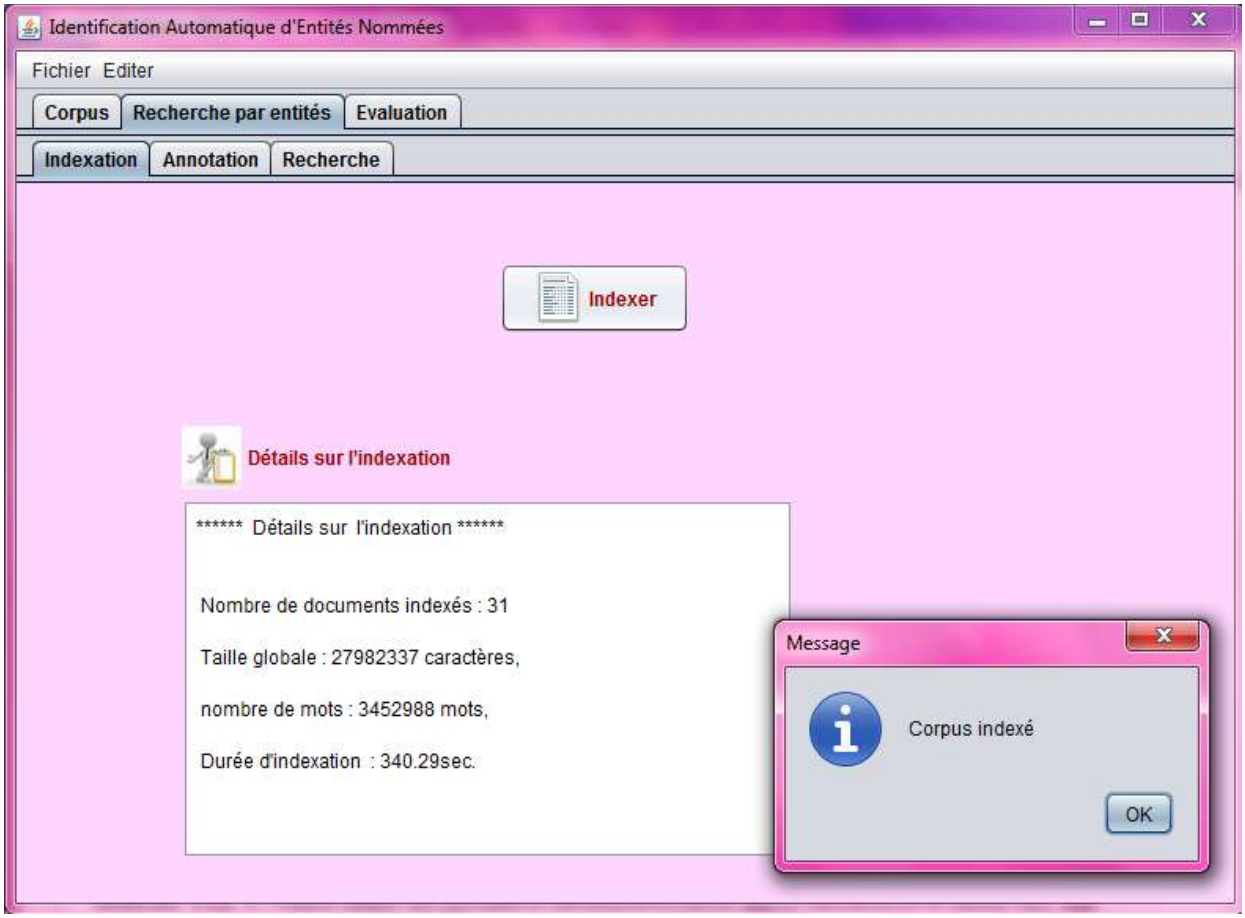


Figure IV.6. Fenêtre d'indexation du corpus

IV.6.2.2. Annotation

En utilisons l'outil Stanford NER pour l'extraction des entités, leurs types. Nous présentons la figure suivante l'interface de confirmation de l'annotation de notre corpus.

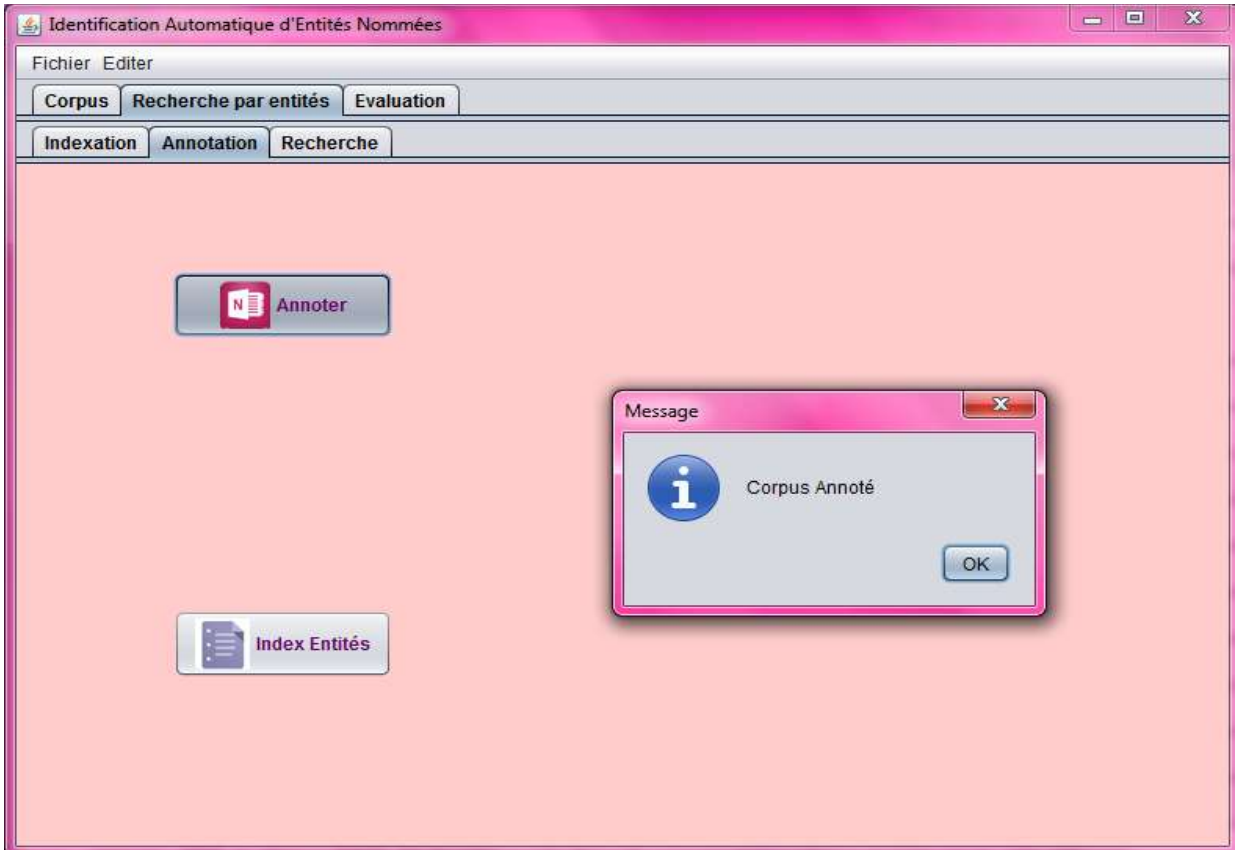


Figure IV.7. Fenêtre de l'annotation du corpus

IV.6.2.3. Index entités

EI (Entités Index) : les entités sont indexées de la même manière que les mots clés. Un index inversé (EI) est construit pour les entités en plus de l'index inversé traditionnel des mots clés (KI). L'index EI retournera une liste qui contient les informations des l'entités.

L'index EI est utilisé pour trouver les documents relatifs aux entités dans le cas de la recherche par entités, il est utilisé également pour trouver les documents relatifs aux entités une fois l'ensemble construit. Il est utilisé aussi dans le cas la diversification par types.

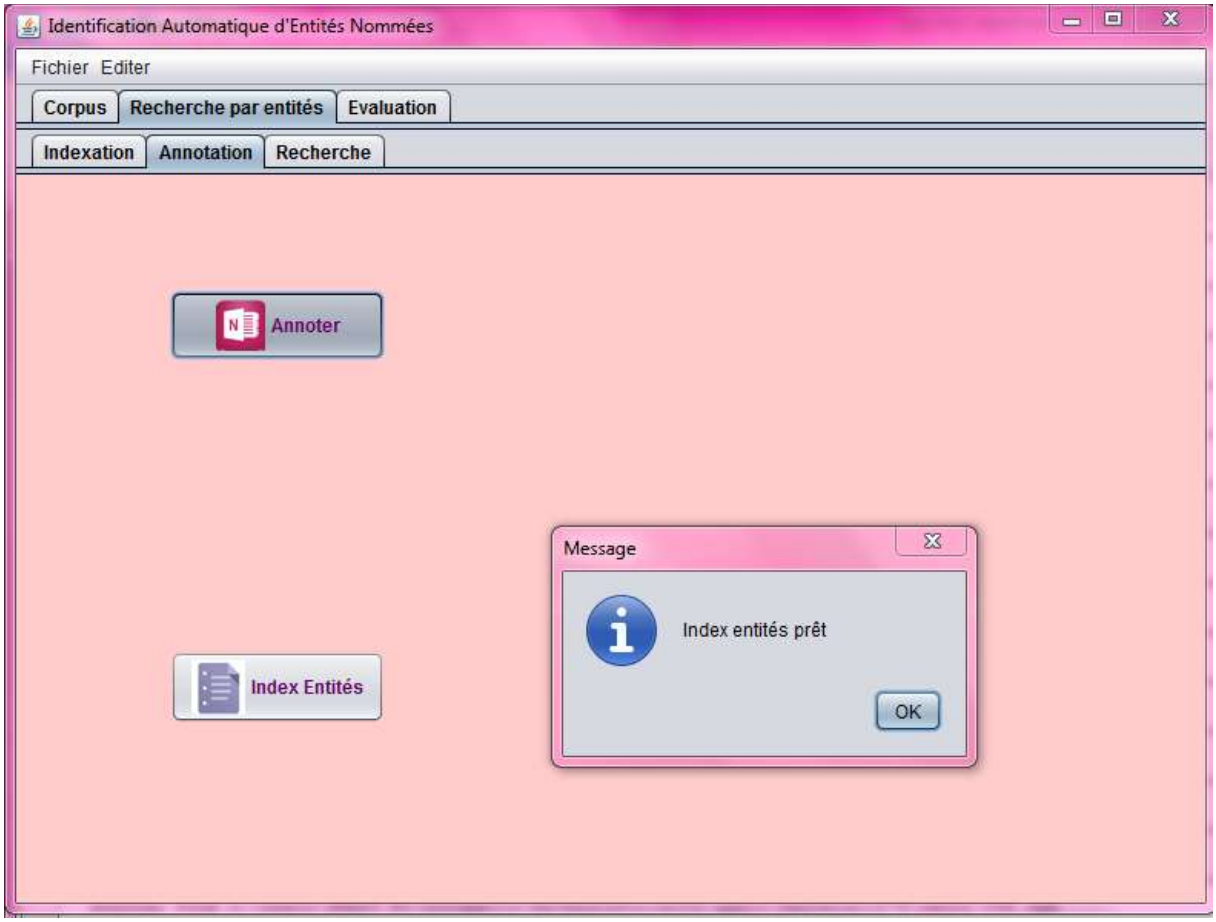
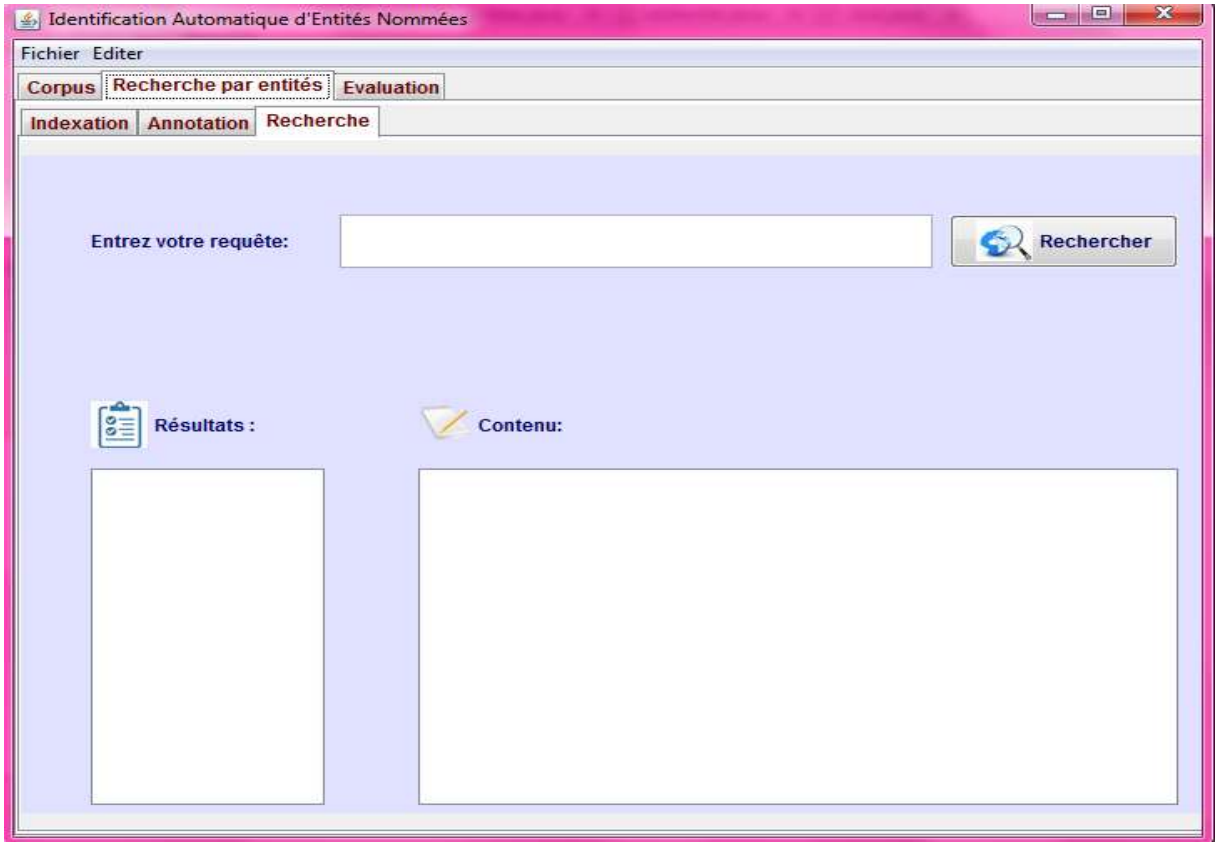


Figure IV.8. Fenêtre de l'index entités

IV.6.2.4. Recherche

L'interface de recherche propose à l'utilisateur de saisir une requête libre.



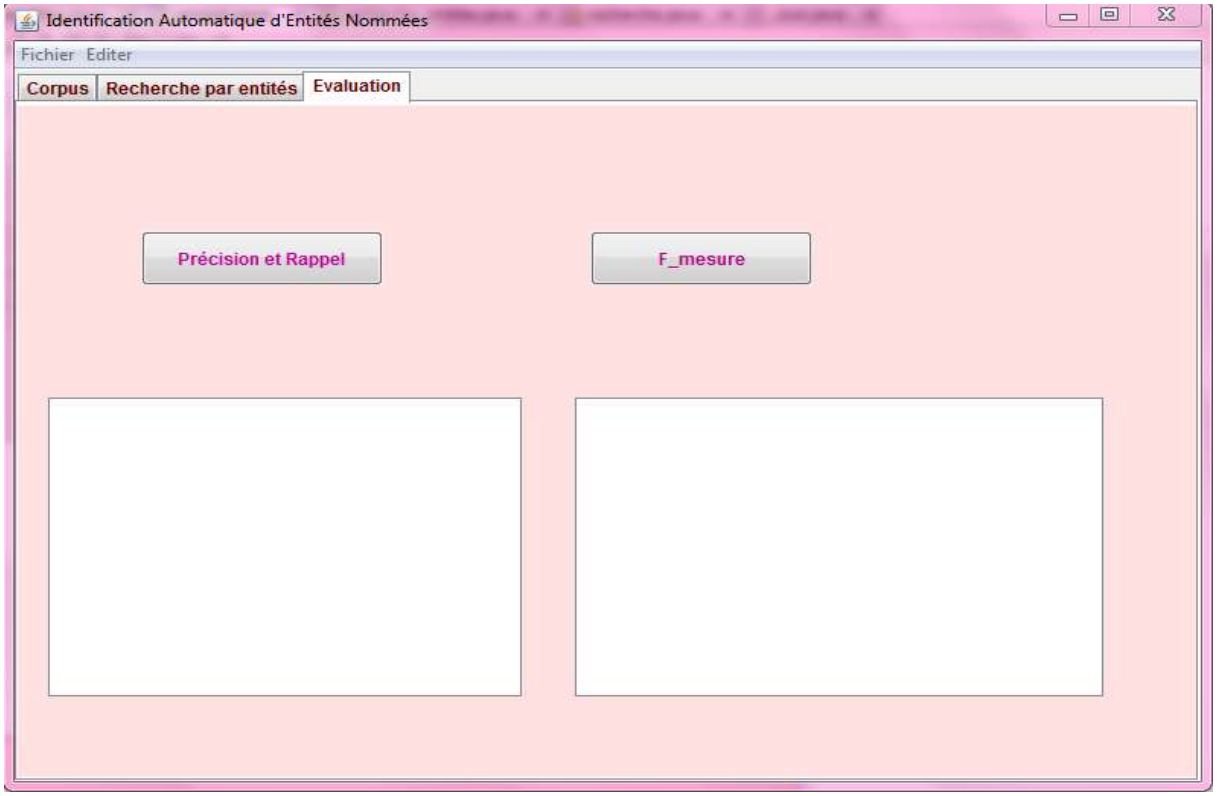
FigureIV.9. L'interface de recherche

- ✓ **Enter votre requête** : pour saisir la requête à rechercher.
- ✓ **Rechercher** : permet de lancer la recherche dans l'index.
- ✓ **Résultats** : Afficher la liste des documents trouvés pertinents par un ordre décroissant par rapport le score et avec la possibilité de les consulter.
- ✓ **Contenu** : pour consulter les documents trouvés pertinents et visualiser le mot recherché dans les documents trouvés.

IV.6.3. Evaluation

IV.6.3.1. Mesures d'évaluation

Dans nos expérimentations, nous nous intéressons en particulier aux mesures suivantes : Rappel, Précision et F-Mesure calculés selon les équations définies dans le **Chapitre 1.**



FigureIV.10. L'interface d'évaluation

IV.7. Conclusion

Dans ce chapitre on a présenté l'implémentation de notre application, l'objectif principal de cette implémentation et de créer un système capable d'identifier des entités nommées.

Conclusion général

La Recherche d'Information a pour objectif de fournir à un utilisateur un accès facile à l'information qui l'intéresse, cette information étant située dans une masse de documents textuels. Afin d'atteindre cet objectif, un système de recherche d'information doit représenter, stocker et organiser l'information puis fournir à l'utilisateur les éléments correspondant au besoin d'information exprimé par sa requête.

Les entités nommées (personnes, lieux, organisations, dates, expressions numériques, marques, fonctions, etc.) sont sollicitées afin de catégoriser, indexer ou, plus généralement, manipuler des contenus.

La reconnaissance d'entités nommées est une tâche dont l'objectif est d'extraire et de typer des éléments informationnels à partir d'un texte donné. Des systèmes de reconnaissance de noms propres à base de ressources linguistiques.
Dans ce travail, Nous avons commencé à définir les concepts de base de la recherche d'information (document, requête, collection de document, …) ensuite nous somme passé à expliquer le fonctionnement de système de recherche d'information tout en incluant les étapes de son processus, puis nous avons cité les modèles de RI (modèle booléen, vectoriel, probabiliste) ensuite nous somme passé à la recherche d'information sur le web on citant les différents types d'outils de recherche sur internet, ensuit le processus de l'extraction d'information avec son principe d'extraction et l'annotation sémantique.

Dans le deuxième chapitre nous avons abordé les entité nommée (définition et quelques exemples) ensuit nous avons expliquer son rôle et leurs différents formes, puis notre intérêt qui il est la reconnaissance et la détection des entités nommées tout en incluant les approche (approche orienté connaissance, approche orientées connaissances, approche hybride) et son techniques (approche symbolique, approche statistique, et approche hybride), puis ont défilé dans l'annotation d'entités nommées et ces éléments l'essentiels, ensuit on définit sont les mesures utilisées dans les évaluations (le rappel, la précision et la F-mesure), enfin, on cite quelques problème et difficultés majeures dans certains domaines(indexation , recherche, question-réponse, …),et on termine le chapitre par une liste des principales campagnes d'évaluations conduites sur notre problématique.

Ensuite, on passant au troisième chapitre dont nous avons présenté la conception de notre système d'identification automatique d'entités nommées qui est basé sur l'indexation et l'annotation et le la recherche On a schématisé les différentes fonctionnalités de ces derniers puis on a passé à la définition de l'ensemble des algorithmes qui représente notre système.

Enfin, on a abordé le quatrième chapitre là on a implémenté notre processus d'identification automatiques d'entités nommées, on présentant ses différents interfaces d'indexation, annotation et la recherche.

Bibliographie

[1] Abbassi Meftah, « Un modèle de reformulation des requêtes pour la recherche d'information sur le Web », mémoire du master, chapitre 01, Les systèmes de recherche d'information.

[2] Rosa Stern, « Identification automatique d'entités pour l'enrichissement de contenus textuels », thèse de doctorat, l'Université Paris 7 Denis Diderot École doctorale de sciences du langage Linguistique théorique, descriptive et automatique, France, 2014.

[3] Saidi Imène, «Contributions aux techniques de recherche d'informations », thèse de doctorat en l'informatique, l'Université d'Oran Ahmed Benbella, Oran – LITIO, 2014.

[4] Souhir Gahbiche Braham, Hélène Bonneau Maynard, François Yvon « Traitement automatique des entités nommées en arabe : détection et traduction », Actes de la conférence conjointe JEP-TALN-RECITAL 2012, volume 2: TALN, pages 487–494, Université Paris- Sud, France, 2013.

[5] Sylvain Goulet, « Technique d'identification d'entités nommées et de classification non-supervisée pour des requêtes de recherche web à l'aide d'informations contenues dans les pages web », Mémoire pour obtention du grade de maitre ès sciences, Faculté des sciences Université de Sherbrooke. Sherbrooke, Quebec, Canada, 2014.

[6] Hammache Arezki, « Recherche d'Information : un modèle de langue combinant mots simples et mots composés », thèse de doctorat en informatique de l'Université Mouloud Mammeri de Tizi-Ouzou, 2010.

[7] Jerry R. Hobbs, « Extraction d'information », Chapitre 21, Université of Southern California Ellen Rilo, Université de l'Utah, Les Etats Unit, 2010.

[8] Vincent Jousse, « Identification nommée du locuteur : exploitation conjointe du signal sonore et de sa transcription », thèse de doctorat en informatique école doctorale 503. sciences et technologies de l'information et mathématiques, université du Mainte, Universités du Nantes, France, 2010.

[9] Fatima Deffaf, « Extraction des entités nommées par projection cross-linguistique et construction de lexique bilingues d'entités nommées pour la traduction automatique statique », mémoire présenté comme exigence partielle de la maitrise en informatique, Université du Québec à Montréal, 2015.

[10] Damien Nouvel, « Reconnaissance des entités nommées par exploration de règles d'annotation Interpréter les marqueurs d'annotation comme instructions de structuration locale », thèse de docteur en Informatique, l'École Doctorale MIPTIS, l'université François Rabelais de Tours, France, 2012.

[12] Meryem Talha, Siham Boulaknadel, Driss Aboutajdine, « Système de reconnaissance des entités nommées amazighes », université Mohammed V-Agdal Rabat 4, Avenue Ibn Battouta, B.P. 1014 RP, 10006 Rabat, 21ème Traitement Automatique des Langues Naturelles, Marseille, France, 2014.

[13] Mohamed Hatmi, « Reconnaissance des entités nommées dans des documents multimodaux », thèse de doctorat en Informatique, Ecole doctorale sciences et technologies de l'information et mathématique, Université de Nantes, France, 2014.

[14] Inès Zribi, Souha Mezghani Hammami, Lamia Hadrich Belguith, « L'apport d'une approche hybride pour la reconnaissance des entités nommées en langue arabe », ANLP Research Group – Laboratoire MIRACL, Faculté des Sciences Economiques et de Gestion de Sfax, Montréal, France, 2010.

[16] Thierry Poibeau, « Du texte brut au web sémantique », LIPN, CNRS et Université Paris 13, France.

[17] Siham Boulaknadel, « Traitement Automatique des Langues et Recherche d'Information en langue arabe dans un domaine de spécialité : Apport des connaissances morphologiques et syntaxiques pour l'indexation », thèse de doctorat en l'informatique, Université de Nantes, École doctorat STIM, France, 2010.

[19] Azeddine Zidouni, « Modèle graphique discriminants pour l'étiquetage de séquences : Application à la reconnaissance d'entités nommées radiophonique», thèse de doctorat en l'informatique, l'Université de la méditerranée, Marseille, 2010.

[20] Ould Hadri Imene Mansouria, «Conception et intégration d'un analyseur morphologique arabe dans un moteur de recherche», mémoire du master en Informatique, Université Abdelhamid Ibn Badis, Mostaganem, 2016.

[21] Kamel Moussaoui, Dhia elhak Feredj, « Conception et développement d'un Outil de recherche sur le web à base d'agent », mémoire du master académique, université Kasdi Merbeh, Ouargla, 2013.

[22] Abd Elkrim. Bouramoul, « recherche d'information contextuelle et sémantique sur le web », thèse de doctorat, université Mentouri Constantine, 2011.

[23] Sabiha Brahimi, Hamza Kouadri, « Amélioration de la précision et du temps de réponse d'un moteur de recherche de texte » mémoire du master académique, Université Kasdi Merbah, Ouargla, 2014.

Webographie

[11] Entités nommées. Technolangue.net. URL http://www.technolangue.net/imprimer.php3?id_article=295. La date de consultation est : 21/12/2016.

[15] URL http://www.ho2s.com/fr/services-web/extraction-de-concepts/ la date de consultation est : 24/12/2016.

[18] URL https://cours-informatique-gratuit.fr/cours/les-moteurs-de-recherche. La date de la consultation est : 12/01/2017.