MINISTERE DE L'ENSEIGNEMENT SUPERIEUR

ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE ABDELHAMID IBN BADIS DE MOSTAGANEM

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

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

Option : Ingénierie des Systèmes d'Information

Approche sémantique pour la résolution d'entité

Réalisé par :

- Nesrine Kréchiche
- Lakhdar Hanchour Haoua

Encadré par :

Mme. Kenniche Ahlem

Année Universitaire : 2018/2019

DEDICACES

A l'âme de mon cher grand père qui m'a toujours conseillé

A mes chers parents, pour tous leurs sacrifices, leur amour, leur tendresse, leur soutien et leurs prières tout au long de mes études.

A mon grand frère Younes, pour son appui et son encouragement et sa confiance en moi.

A mon adoré petit frère Mohamed.

A toute ma famille pour leur soutien tout au long de mon parcours universitaire.

A tous mes professeurs et surtout à mon encadreur.

Je dédie ce travail...

Nesrine

Je dédie ce travail :

A l'âme de mon cher père qui était la cause dont je suis maintenant.

A ma chère maman et ma sœur Fatiha qui m'ont encouragé.

A mon frère Ghali et sa femme Souhila.

A mon frère Hamidou.

Haouaa

DEDICACES

REMERCIEMENTS

Tout d'abord et avant toute chose, on remercie Dieu le tout puissant de nous avoir donné la force, la foi et la volonté de mener ce travail terme.

C'est avec un immense plaisir qu'on rédige cette page à travers laquelle on souhaite exprimer notre profonde gratitude aux personnes qui ont contribué à l'accomplissement de cette thèse.

On remercie notre encadreur, Madame Kenniche Ahlem, Professeur de l'université Abd El Hamid Ben Badis avec qui on a eu l'honneur de travailler, de nous avoir guidé et orienté d'un bout à l'autre de ce chemin. On la remercie pour avoir accordé du temps à une lecture attentive et détaillée de notre manuscrit ainsi que pour ses remarques encourageantes et constructives. On voudrait dire merci tout spécialement pour sa patience, sa disponibilité et sa compréhension.

On tient ensuite à remercier tous les enseignants d'informatique de l'université de Mostaganem surtout Madame Bahnes pour sa précieuse aide.

Nos familles … On croit qu'on n'aura pas assez de mots pour vous remercier et vous dire ce que vous représentez pour nous. Vous avez cru en nous, vous êtes toujours là à nos côtés. Nous sommes fières de chacun d'entre vous.

La gaieté de mes amies « Louiza, Souhila, Chaimaa, Amira, Kenza, Kheira », leur présence, on ne vous remerciera jamais assez.

On remercie toutes les personnes qu'on n'a pas cité, bien qu'elles aient contribué, de loin ou de près, pour l'accomplissement de ce travail.

A tous MERCI.

RESUME

Vue le très grand nombre de pages et de liens, il est devenu primordial de disposer des systèmes de recherche d'information efficaces et performants pour retrouver des informations pertinentes en un temps opportun.

Dans ce contexte, le problème n'est plus la disponibilité de l'information, mais la capacité de sélectionner celle qui répond le mieux au besoin d'un utilisateur.

Ce mémoire traite la recherche d'information sémantique basé sur des annotations d'entités nommées en évaluant les deux extrêmes de la RI : Document et Requête par leur entités avec leurs types et leurs mots-clés. L'objectif est de retourner un résultat satisfaisant et qui répond pertinemment à la requête d'un utilisateur.

Mots-Clés : Recherche d'information (RI), Document, Requête, Entité Nommée (EN), Reconnaissance des entités nommées (REN), Traitement Automatique des Langues (TAL).

SOMMAIRE

SOMMAIRE

SOMMAIRE

LISTES DES FIGURES

INTRODUCTION GENERALE

Avec l'avènement d'Internet, la recherche d'information est devenue, un domaine important dans la communauté de la recherche scientifique. Aujourd'hui, la recherche d'information est un champ transdisciplinaire qui peut être étudié par plusieurs disciplines ou approches qui permettent de trouver des solutions pour améliorer son efficacité. Elle permet le stockage et la représentation de l'information d'une part, l'analyse et la satisfaction d'un besoin d'autre part.

Le développement des données disponible sur internet a considérablement changé le domaine du traitement des langues. Il n'y a pas longtemps ces systèmes traitaient des données mais aujourd'hui, ils doivent faire face à des déluges de documents variés.

Dans ce mémoire, nous nous intéressons à une des taches de la recherche d'information qui est la reconnaissance des entités nommées, Elles correspondent généralement à l'ensemble des noms propres (noms de personnes, noms de lieu, dates…) et sont actuellement bien reconnues par les systèmes automatiques.

Problématique

Bien que simple et intuitive, la recherche d'information telle qu'elle est largement utilisée aujourd'hui n'est pas toujours adaptée à certains besoins. Dans certains cas, les utilisateurs ne cherchent pas une liste ordonnée de documents mais les informations que ceux-ci contiennent, ceci revient à trouver des entités à la place de documents. Dans ce contexte de recherche les utilisateurs pourraient être intéressés par découvrir les documents contenants les entités relatives à une maladie particulière (comme les symptômes d'une maladie).

Nous considérons alors le problème de la recherche des entités et des documents les contenants en réponse aux requêtes d'utilisateurs.

Notre idée est de construire un système de recherche d'entités en nous basant sur des annotations, pour prendre en charge des requêtes construites par une ou plusieurs entités (recherche par entités) ainsi que les requêtes de mots clés (recherche d'entités par mots clé) et de retourner des entités relatives aux requêtes avec les documents résultats pour chaque entité.

Trouver des entités à la place des documents dans le web est un axe de recherche récent de la recherche d'information.

Structure du mémoire

Ce mémoire est organisé en quatre parties :

La première partie : c'est une présentation de l'état de l'art relatif à la tâche de la recherche d'information (RI). Nous commençons par des définitions générales, des notions de bases de la recherche d'information, le traitement des langues (TAL), les systèmes de recherche.

INTRODUCTION GENERALE

La deuxième partie : met l'accent sur les entités nommées en les définissant, nous présentons par la suite leurs types, leurs rôles, leurs reconnaissances et leurs extractions avec ces modèles.

La troisième partie : est une conceptualisation de notre système, premièrement nous avons défini le logiciel GATE et ses composants, deuxièmement nous avons schématiser notre démarche pour réaliser notre système.

La quatrième partie : est consacrée pour l'implémentation, dont nous avons défini le langage et l'environnement de développement du système en les illustrant avec des captures d'écran de notre système réalisé.

Nous terminons ce mémoire avec une conclusion générale et une bibliographie.

CHAPITRE 1 :
RECHERCHE D'INFORMATION

CHAPITRE 1 : RECHERCHE D'INFORMATION

I.1 Introduction :

« Alors qu'il y a quelques siècles, les gens avaient du mal à accéder à l'information. Aujourd'hui, beaucoup s'efforcent d'éliminer l'information non pertinente qui leur parvient par différents canaux.» [Bukley, Berners Lee].

La croissance continue du volume de données (texte, image, vidéo) présentée sous différents formats, ainsi que l'apparition de disques offrant de gigantesques espaces de stockage ont imposé des mécanismes pour gérer cette masse d'informations. Ce besoin a marqué la naissance du domaine de la « Recherches d'Information ».

Dans ce chapitre nous allons aborder le concept de recherche d'information (RI), ses définitions, quelques concepts de base, le traitement automatique des langues (TAL), TF/IDF et les moteurs de recherche.

I.2 Définition de recherche d'information :

Pendant ces dernières années plusieurs définitions ont été proposées parmi eux :

I.2.1 Définition 1 :

La recherche d'informations est une activité dont la finalité est de localiser et de délivrer des granules documentaires à un utilisateur en fonction de son besoin en information [1].

I.2.2 Définition 2 :

Recherche d'information (RI) vise à retrouver des documents répondants à un besoin informationnel (thème) spécifié par une requête [2].

I.2.3 Définition 3 :

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 [3].

Et toute ces définitions partagent l'idée que la recherche d'information a pour objet d'extraire d'un document ou d'un ensemble de documents, les informations pertinentes qui reflètent un besoin d'information [1].

I.3 Concepts de base de la recherche d'information :

La RI se définie par l'identification de documents qui satisfassent le mieux le besoin en informations d'un utilisateur, ces documents doivent être trouvés parmi une large collection de documents.

Le défi est de pouvoir, parmi le volume important de documents disponibles trouver ceux qui correspondent aux mieux à l'attente de l'utilisateur.

CHAPITRE 1 : RECHERCHE D'INFORMATION

Nous présentons en ce qui suit, quelques concepts nécessaires à la recherche d'information :

I.3.1 Document :

Ensemble composé d'un contenu, d'une structure logique, d'attributs de présentation permettant sa représentation, exploitable par une machine afin de restituer une version intelligible pour l'être humain. Le document peut être créé à l'état natif ou obtenu par un processus de transformation d'un document physique.

I.3.2 Document pertinent :

Un document pertinent est un document étendu dans le besoin informationnel de l'utilisateur, validé, clarifié, actuel, a un caractère tangible et une bonne source en qualité et réputation.

I.3.3 Fond documentaire :

Est une collection de documents constituant l'ensemble des informations exploitables et accessibles dans la recherche ou un ensemble cohérent de document, établi en vue d'un usage précis, faisant l'objet d'une gestion. Chacun des objets qui la composent a plus de valeur dans l'entité collective qu'il n'en aurait individuellement.

I.3.4 Requête :

Est l'interrogation et la recherche de l'utilisateur exprimant ses besoins, elle est parfois claire comme « Université Abd El Hamid Ben Badis » et parfois non clair comme par exemple « Abd El Hamid Ben Badis ».

I.3.5 Index :

Désigne en général une structure chargée d'ordonner et de trier les documents (e-book, pdf, image, adresse url…) afin de pouvoir les retrouver plus rapidement. L'un des index les plus connus est sans doute celui utilisé par Google pour le référencement des sites web et l'affichage des résultats dans son moteur de recherche [4].

I.3.6 L'indexation :

L'indexation est un processus primordial permettant de construire un ensemble d'éléments « clés » permettant de caractériser le contenu d'un document ou retrouver ce document en réponse à une requête [5].

I.3.7 Besoin d'information :

Un besoin d'information correspond à une sensation de manque de connaissance d'un individu dans une situation l'engageant dans une activité de recherche d'information.

I.3.8 Modèle de recherche :

Un modèle de recherche d'information propose une manière de représenter les requêtes et les documents ainsi qu'une fonction de correspondance qui associe des scores aux couples requête-document permettant ainsi de trier les documents en fonction de la requête.

Un modèle de RI est défini par un quadruplet {D, Q, F, R (q, d)} où [6]:

- D est l'ensemble des documents.
- Q est l'ensemble des requêtes.
- F est le schéma du modèle théorique de représentation des documents et des requêtes.
- R (q, d) est la fonction de pertinence du document et la requête q.

Nous allons décrire ici quelques modèles de la recherche d'information (RI) :

I.3.8.1 Modèle booléen :

Le modèle booléen est le modèle le plus simple des modèles de RI, il est basé sur la théorie des ensembles et l'algèbre booléenne. Il propose une représentation de la requête sous forme d'une expression logique dont les termes d'indexation sont reliés par les connecteurs logiques (ET (∧), OU (∨) et NON (¬)) [7], qu'ils effectuent des opérations d'union, d'insertion et de différence entre les ensembles de résultat associés à chaque terme [1].

Exemple :

L'utilisateur tape une requête « q » qui a des mots-clés présentés comme suit :

q = programmation ∧ langage ∧ (C ∨ Java).

Donc q = [programmation ∧ langage ∧ C] ∨ [programmation ∧ langage ∧ Java].

Pour trois documents différents, le résultat de la recherche était :

	Programmation	Langage	C	Java
Document 1	3(1)	2(1)	4(1)	0(0)
Document 2	5(1)	1(1)	0(0)	0(0)
Document 3	0(0)	0(0)	0(0)	3(1)

Dont les résultats du tableau représentent le nombre d'apparition du mot dans le document.

Finalement le score de pertinence a deux possibilités :

- RSV (dj, q) = 1 si le terme existe dans le document.
- RSV (dj, q) = 0 sinon.

I.1.3.8.2 Modèle probabiliste :

La modélisation probabiliste dans le domaine de recherche d'information influencé par la théorie mathématique des probabilités, consiste à utiliser un modèle qui classe les documents dans un ordre décroissant de leurs probabilités de pertinence à un besoin d'information d'un utilisateur.

L'idée de ce principe est de retrouver des documents qui ont en même temps une forte probabilité d'être pertinents et une faible probabilité d'être non pertinents, c'est-à-dire de classer l'ensemble des documents en deux classes notamment, la classe des documents pertinents notée « R » et la classe des documents non pertinents notée « NR » à un besoin d'utilisateur [8].

Les modèles probabilistes sont importants parce qu'ils représentent une des tentatives les plus significatives pour donner une base théorique solide à la RI.

I.1.3.8.3 Modèle vectoriel :

Le modèle vectoriel représente un document ainsi qu'une requête par un vecteur dans un espace dont chaque dimension correspond à un descripteur atomique. Chaque coordonnée dans cet espace dénoté l'importance du descripteur dans le document considéré. Le traitement d'une requête est alors basé sur la comparaison des vecteurs documents et requêtes [9].

La figure suivante montre une représentation sur ce modèle :

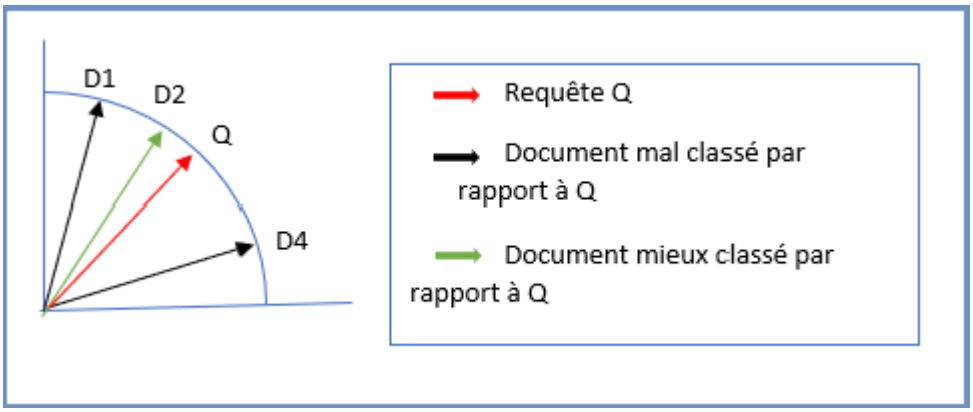


Figure 1.1 : Représentation vectorielle.

La recherche d'information, dans la mesure où elle travaille aussi sur des textes, s'apparente au TAL. Leurs liens sont anciens et leurs frontières sont perméables :

I.4 Définition de TAL :

Le traitement automatique des langues (TAL) est une discipline à la frontière de la linguistique, de l'informatique et de l'intelligence artificielle. Elle concerne la conception de systèmes et techniques informatiques permettant de manipuler le langage humain dans tous les aspects [10].

I.5 Les grands domaines du traitement automatique des langues :

Nous présentant ici les grands domaines du TAL, en nous appuyant sur un découpage méthodologique classique dans le domaine et en linguistique [2] :

- **La morphologie :** Concerne l'étude de la formation des mots et de leurs variations de formes.
- **La syntaxe :** S'intéresse à l'agencement des mots et à leurs relations structurelles dans un énoncé.
- **La sémantique :** Se consacre au sens de l'énoncé.
- **La pragmatique :** Prend en compte le contexte de l'énoncé.

I.6 Les niveaux de traitement du Tal sur les documents :

Nous introduisons ici les différents niveaux de traitement nécessaires (voir figure.1.2.) pour parvenir à une compréhension complète d'un énoncé en langage naturel. Du point de vue d'un ingénieur, ces niveaux correspondent à des modules qu'il faudrait développer et faire coopérer dans le cadre d'une application complète de traitement de la langue [2] :

- **Segmentation :** Diviser le texte en unités lexicales (mots).
- **Traitement lexical :** Identifier les composants lexicaux et leurs propriétés.
- **Traitement syntaxique :** Identifier des groupes (constituants) de plus haut niveau et les relations de dominance qu'ils entretiennent entre eux.
- **Traitement sémantique :** Construire une représentation du sens de cet énoncé, en associant à chaque concept évoqué un objet ou une action dans un monde de référence (réel ou imaginaire).
- **Traitement pragmatique :** Identifier enfin la fonction de l'énoncé dans le contexte particulier de la situation dans lequel il a été produit.

I.7 Exemple de TAL :

On montre sur cette figure une phrase interprétée avec TAL avec ses niveaux, cette phrase est prise d'une recette d'un dessert : « Coupez les fruits en morceaux » :

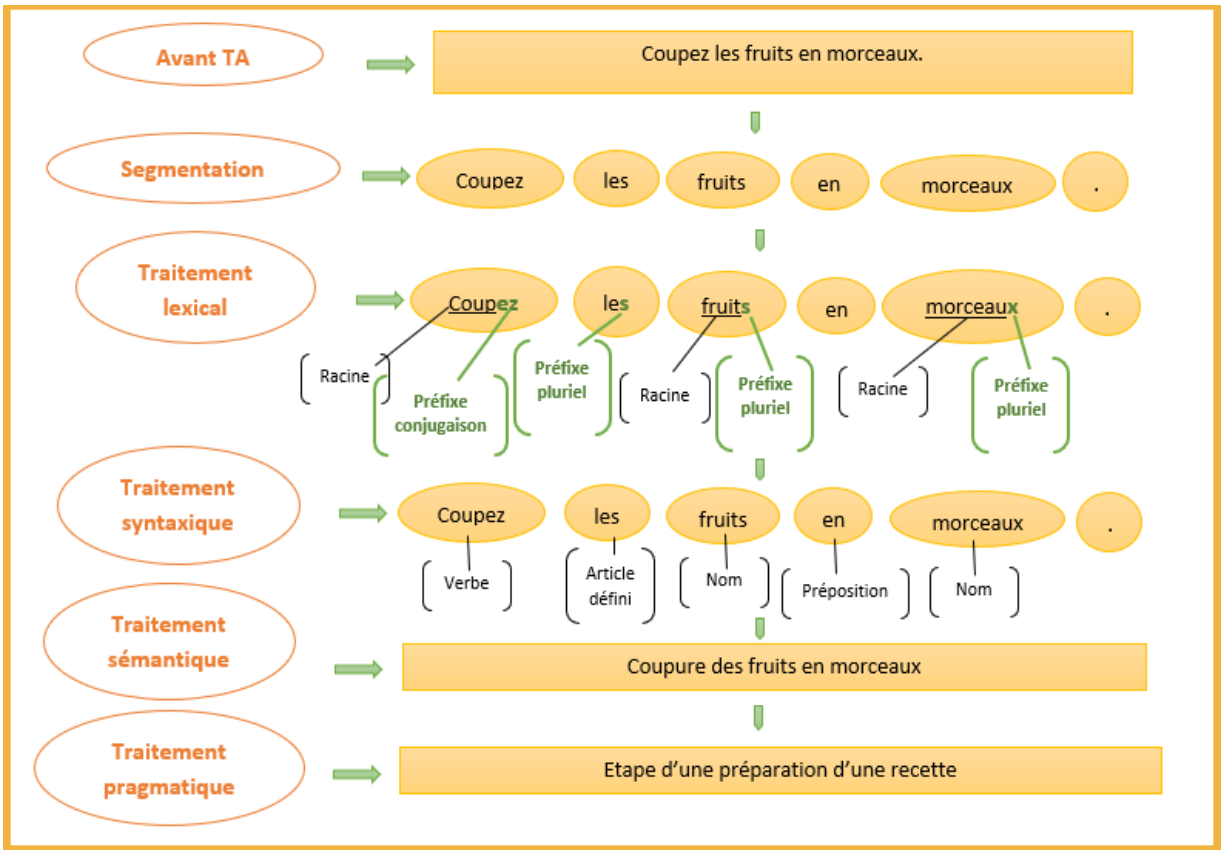


Figure 1.2: Le processus du traitement automatique des langues (TAL).

I.8 Systèmes de recherches d'informations (SRI) :

Ce sont les systèmes qui font la recherche d'information et qui ont pour but de mettre en correspondance une représentation du besoin de l'utilisateur (requête) avec une représentation du contenu des documents (fiche ou enregistrement) au moyen d'une fonction de comparaison (ou de correspondance) et ça en retrouvant une information pertinente dans un espace diversifié et de taille considérable [1].

I.9 Processus du système de recherche d'information :

La recherche d'information est 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. Comme le montre la figure.1.3. Ceci implique en général ces processus :

CHAPITRE 1 : RECHERCHE D'INFORMATION

- L'utilisateur a besoin d'information.
- Poser une question (requête).
- Traitement de la requête (analyse, compréhension et représentation).
- Indexation des documents.
- Représentation des documents.
- Comparaison entre les requêtes et les documents (appariement).
- Documents trouvés
- Jugement de résultat (utilisateur satisfait ou non satisfait).
- Reformulation de requête ou cas ou l'utilisateur est non satisfait.

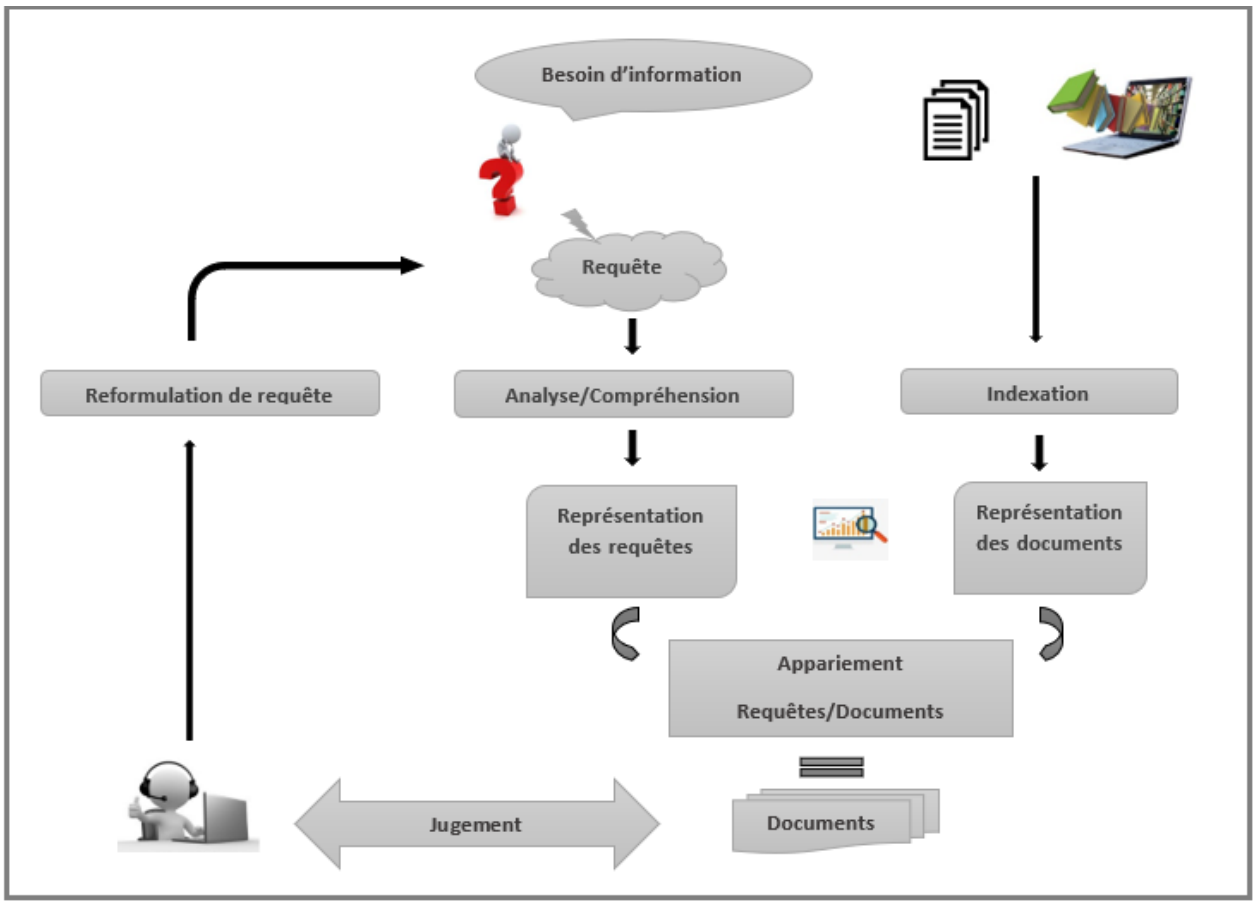


Figure1.3 : Processus du système de recherche d'information

I.10 Explication du processus de SRI :

Le premier processus est lié au facteur cognitif humain, où l'utilisateur a besoin d'information pour accomplir ses prérequis et ses connaissances ou pour acquérir une nouvelle

connaissance pour lui. Pendant ce processus, l'utilisateur parfois ne connait pas vraiment ses besoins ou il est incapable de les définir nettement.

Donc, à partir de cet état cognitif mal défini, il tente de s'exprimer dans le langage utilisé par le système en produisant ce qu'on l'appelle « Requête ».

Puis, le système de recherche d'information traite la requête en l'analysant en prenant en compte des difficultés dues à l'ambiguïté du langage naturel. Il fait aussi l'indexation des documents et leur représentation.

Ensuite, il compare les représentations des requêtes avec celles des documents et présente enfin des solutions de façon compréhensible. Ou cas où l'utilisateur n'est pas satisfait, il reformule de nouveau sa requête.

I.11 Définition de TF/IDF :

« Terme Frequency / Inverse Document Frequency » est le résultat d'un calcul algorithmique de moteur de recherche, permettant d'obtenir un poids, une évaluation de la pertinence d'un document par rapport à un terme, en tenant compte de deux facteurs : la fréquence de ce mot dans le document (TF) et le nombre de documents contenant ce mot (IDF) dans le corpus étudié [11].

I.12 Exemple de TF/IDF :

La figure.1.4. Illustre un exemple de comparaison entre deux documents selon leurs pertinences et cela par rapport à leur nombre de mots et le nombre d'occurrence des mots clés :

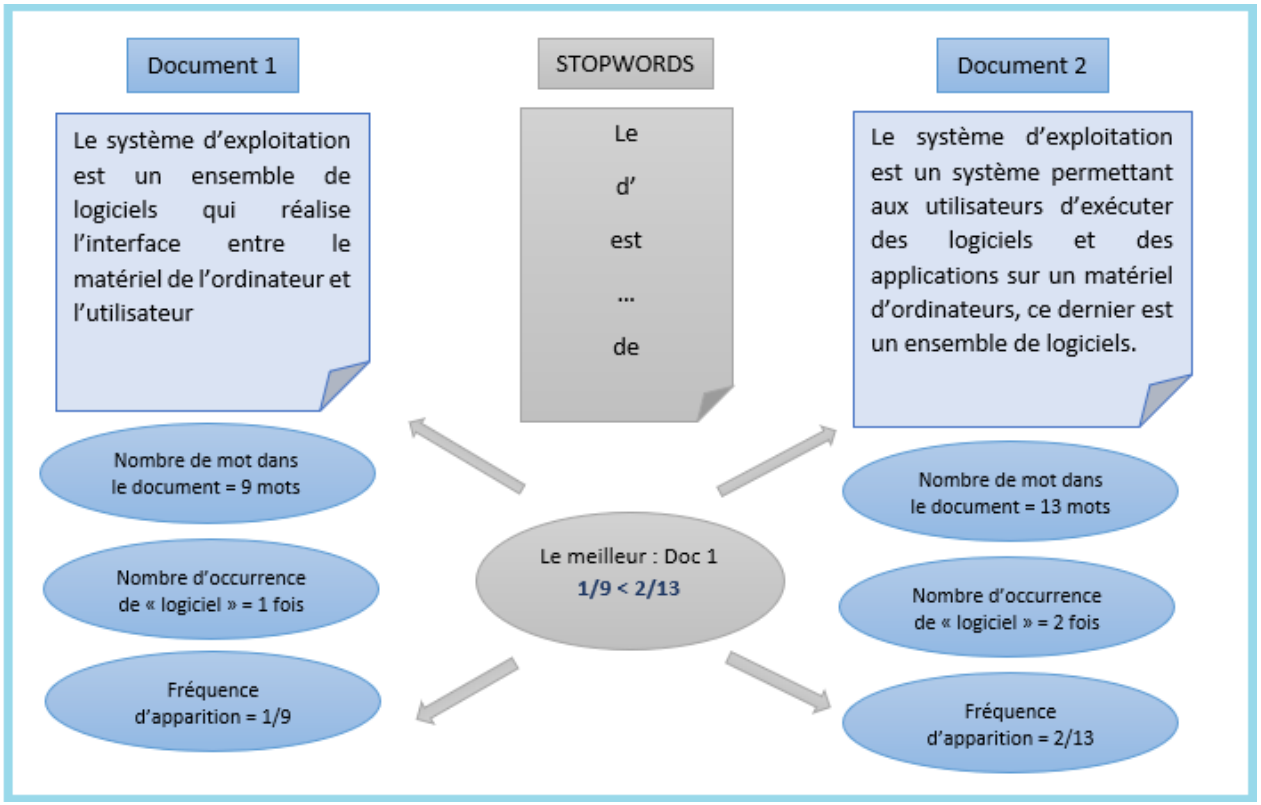


Figure 1.4: Exemple de documents traité par TF/IDF

I.13 Moteur de recherche :

Un moteur de recherche est un logiciel ou un robot de recherche sur internet appelé spider, qui permet de trouver des ressources (sites Web, images, vidéos, fichiers), en parcourant Internet à intervalle régulier et de façon automatique.

Le moteur de recherche suit les liens de millions de pages web, localise en permanence de nouvelles adresses et indexe le contenu dans des gigantesques bases de données. Les utilisateurs interrogent ces bases de données à l'aide des requêtes [12].

Il existe plusieurs moteurs de recherche sur le web, on site parmi eux :

I.13.1 Google :

Est le moteur de recherche qui a donné son nom à la société Google, le plus utilisé au monde, créé en 1998. Son principe de fonctionnement est basé sur le PageRank c'est-à-dire lorsqu'un document est pointé par de nombreux liens son PageRank augmente.

Ce système donne une indication sur la popularité du document parmi les ressources du web, il est très apprécié pour sa rapidité et sa sobriété [12].

I.13.2 Yahoo :

Moteur concurrent de Google (12% des requêtes dans le monde), il offre sensiblement la même qualité de réponse lors des requêtes simples et aussi un index presque conséquent, mais est moins pertinent dans le cas d'une requête complexe. L'index est sensiblement de la même taille que Google [13].

I.13.3 Exalead :

Est un moteur de recherche conçu en France, basé sur la spécificité du langage français et fonctionnant sur le clustering pour générer des termes associés, mais aussi pour catégoriser à partir d'une liste définie (index de 8 milliards de pages). Il permet de prévisualiser les pages web grâce à des vignettes et des fonctions avancées sont proposées pour affiner la recherche (termes associés, type de ressources, langue, annuaire…). Il permet de créer et gérer une sélection de sites favoris grâce à l'option « Ajouter aux raccourcis » (portail personnalisé du type Netvibes) [13].

I.14 Conclusion :

Dans ce chapitre, nous avons abordé le domaine de la recherche informationnelle (RI) dans le web, nous avons détaillé le processus de recherche d'information ainsi que présenté quelques exemples sur des moteurs de recherche, aussi nous avons présenté le domaine du TAL qui est une étape nécessaire pour la construction d'un SRI efficace. Dans le prochain chapitre, nous entamons les entités nommées.

CHAPITRE 2 :
ENTITEE NOMMEE

CHAPITRE 2 : ENTITEE NOMMEE

II.1 Introduction :

La tâche d'étiquetage des entités nommées a reçu une attention considérable au sein de la communauté TAL depuis les années 90. Une des tâches partagées de la série de conférences MUC (Message Understanding Conference) avait pour objectif de reconnaître les entités nommées. Dans ce chapitre nous allons aborder le concept des Entité nommées (EN).

II.2 Définition des entités nommées :

II.2.1 Définition 1 :

Les entités nommées (ENs) désignent l'ensemble des noms de personnes, de lieux, d'organisations, etc. contenues dans un texte. On ajoute souvent à ces éléments les dates, les quantités et d'autres données. Par extension, les entités désignent parfois les éléments de base pour une tâche donnée. Ces séquences référentielles sont primordiales pour beaucoup d'applications linguistiques, que ce soit la recherche ou l'extraction d'information, la traduction automatique ou la compréhension de texte [14].

II.2.2 Définition 2 :

Une EN est une appellation générique pour la catégorisation d'un certain nombre d'objets textuels rencontrés dans un document. Les ENs incluent traditionnellement quatre grandes classes : les noms, les quantités, les dates, les durées [15].

II.2.3 Définition 3 :

Les EN sont des unités lexicales particulières, c'est-à-dire, des objets textuels (un mot ou un groupe de mots) "catégorisables" dans des classes telles que noms de personnes, noms d'organisations ou d'entreprises, noms de lieux, quantités, dates, etc. [16]

En somme, une EN peut donc être définie comme une unité linguistique (syntagme), identifiable de façon unique dans un contexte précis et qui renvoie à un objet du monde réel. [14]

Le schéma suivant montre en quelque sorte la hiérarchie des entités [17] :

CHAPITRE 2 : ENTITEE NOMMEE

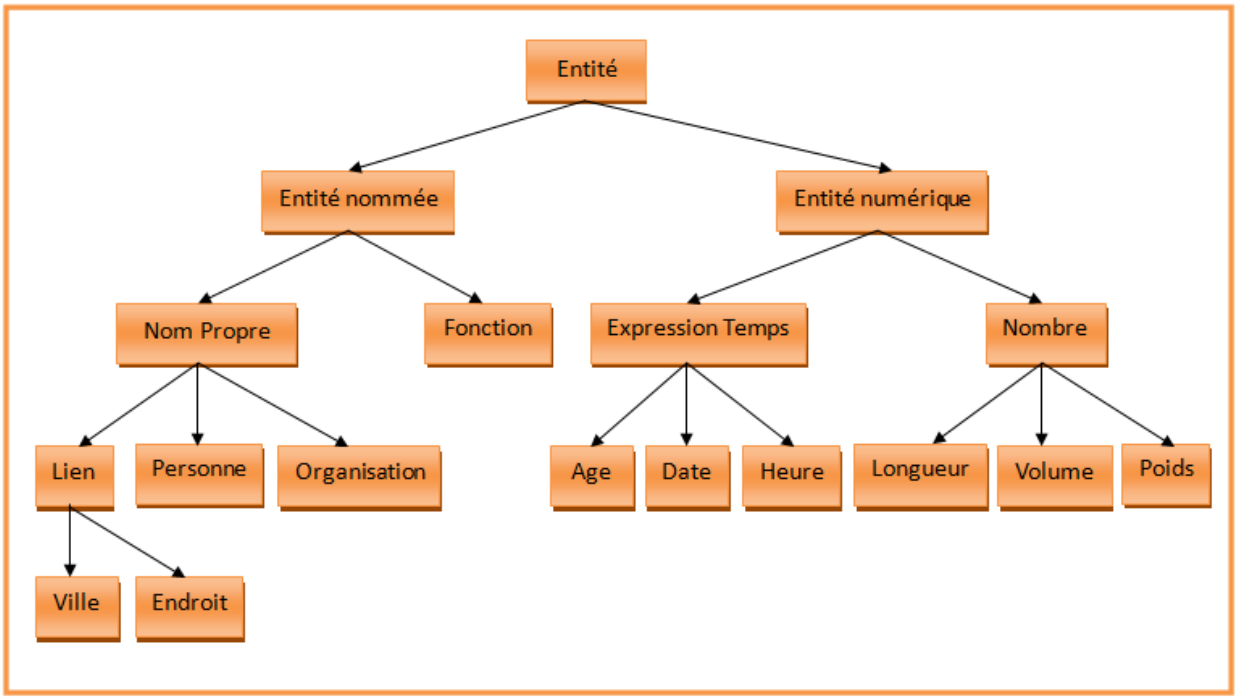


Figure 2.1 : Hiérarchies de types d'entités

II.3 Les types des entités nommées :

Nous traiterons les EN en les représentant en deux principales catégories : les EN «Simple» et les EN «complexes»[14] :

II.3.1 Les EN Simples :

- Une personne (ex : « Bill Gates », « Mark Zuckerberg »).
- Une organisation (ex : « UNICEF », « Google »).
- Un lieu (ex : « Place des Martyrs », « Alger »).
- Une date (ex : « 2019 », « hiver »).

II.3.2 Les EN complexes :

Dans cet exemple, le film est traité comme une EN complexe. Le titre est un texte, le réalisateur est une EN élémentaire de type personne, et les catégories sont représentées par un ensemble de textes :

- Titre : « Matrix » ;
- Date de sortie : « 31 Mars 1999 » ;
- Réalisateur : « Lana Wachowski » et « Lilly Wachowski » ;
- Catégories : « Fantasy », « Science-fiction ».

CHAPITRE 2 : ENTITEE NOMMEE

II.4 Rôle de l'entité nommée :

Les ENs présentent plusieurs avantages dans plusieurs domaines actuels, ce sont la base de tout texte ou document d'où il est très important de passer par eux dans la recherche en TALN (Traitement Automatique des Langues Naturelles), dans le développement des systèmes Questions/Réponses, les résumés automatiques, la recherche d'information (RI), la traduction automatique (TA), et même dans le Web sémantique (WS) [18].

II.5 Reconnaissance des entités nommées :

II.5.1 Définition 1 :

La reconnaissance des ENs (RENs) est une sous-tâche de l'extraction d'informations qui prend en entrée un bloc de texte non annoté et produit un bloc de texte annoté contenant les entités nommées trouvées. Chaque entité reçoit une étiquette en fonction de son type sémantique [16].

II.5.2 Définition 2 :

La REN est la tâche de rechercher des termes qui correspondent à des ENs dans un texte [18], La tâche de reconnaissance d'entités nommées se décline en deux sous traitements le premier est l'identification de ces unités dans un texte, le deuxième est la catégorisation en fonction des types de classes prédéfinis dans la tâche. Par exemple, le mot **Nelson Mandela** peut avoir plusieurs sens, comme le montre l'exemple suivant [20] :

CHAPITRE 2 : ENTITEE NOMMEE

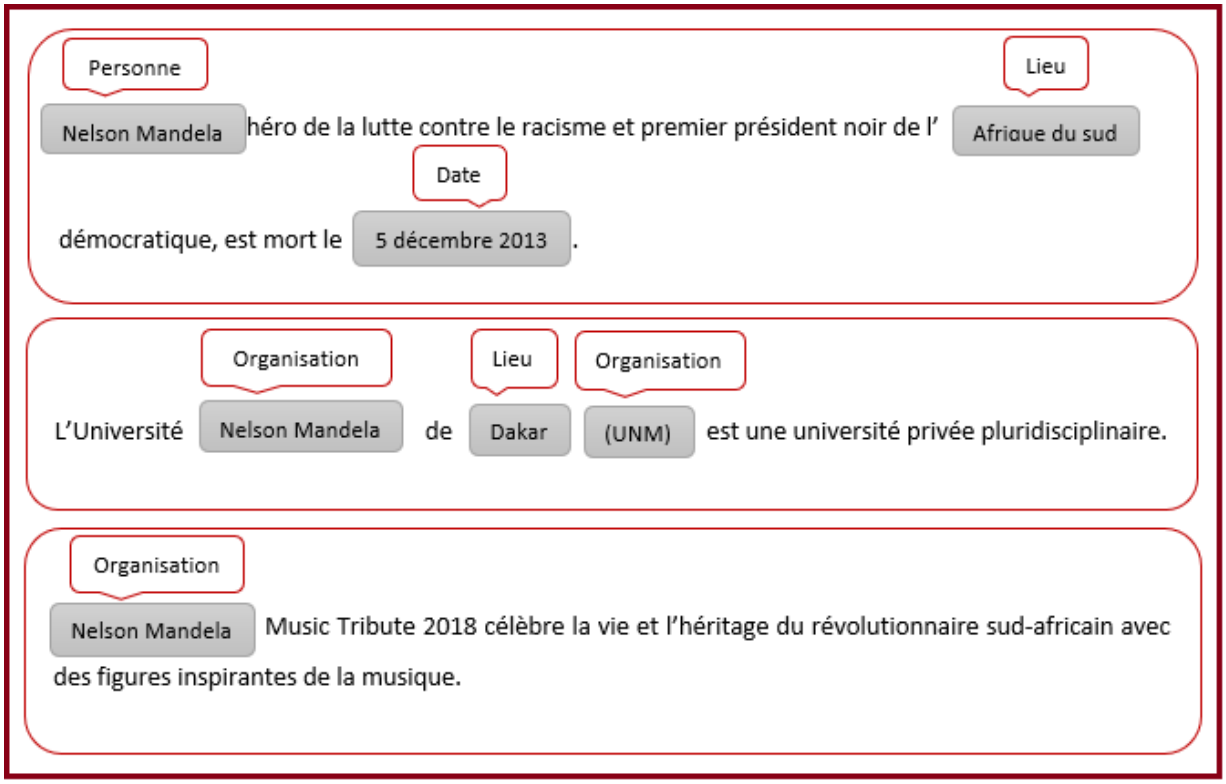


Figure 2.2 : Exemples d'une entité nommée catégorisée selon son contexte.

Cette tâche est réalisée par l'une des approches d'extraction des ENs [18].

II.5.3 Extraction des entités nommées

En TALN, il y trois approches très utilisées qui sont l'approche symbolique ou linguistiques (à base de règles), l'approche statistique (ou à base d'apprentissage) et l'approche hybride ont classifié les systèmes d'extraction des ENs en trois classes selon l'utilisation de l'une de ces trois approches [19].

Depuis quelques années, la recherche dans le domaine de la REN n'arrêtait pas d'évoluer où différentes méthodes sont apparues [16] :

II.5.3.1 Approches symboliques :

Ce sont des approches linguistiques qui se basent sur des règles génériques (règles contextuelles) écrites à la main (patrons d'extraction) [16], elle est utilisée par la majorité des systèmes de reconnaissance d'entités nommées [20] :

Exemples [21] :

- **Noms propres :** le mot commence par une majuscule ('Zighoud') ;
- **Personnes :** le premier token est un prénom ('Youcef Zighoud') ;
- **Dates :** le premier et le dernier token sont composés de chiffres ('5 juillet 1962') ;

- **Organisations :** le dernier token est “I.S.O” ou “ISO” (International Organization for Standardization) ;
- **Lieux :** contient “sur” ou “en” suivi d’un nom (‘Gare du puy-en-velay’).

Nemesis [22] est un système qui permet la délimitation et la catégorisation des entités nommées développé pour le français et pour un texte bien formé, il est architecturé comme suivant (voir figure 2.4) :

- **Prétraitement lexical :** Il s’agit d’un processus de segmentation du texte en phrases et en formes, puis association des sigles et de leur forme.
- **Projection des lexiques :** Elément composant permet de réparer les lexiques selon les catégories dont ils sont utilisés
- **Application des règles :** Une fois la projection des lexiques est réalisée, les règles de réécriture sont appliquées, pour permettre une REN.

Ces règles de réécriture permettent l’annotation du texte par des balises identifiant les entités nommées. Elles sont basées sur des étiquettes sémantiques référant à une forme capitalisée ou à une forme appartenant à un lexique [28].

L’emploi de ces règles peuvent être fait automatiquement et correctement avec des outils d’extraction d’informations parmi eux Open Calais et Gate. Ces outils-là traitent plusieurs types de documents en les annotant pour avoir à la fin une REN convenable.

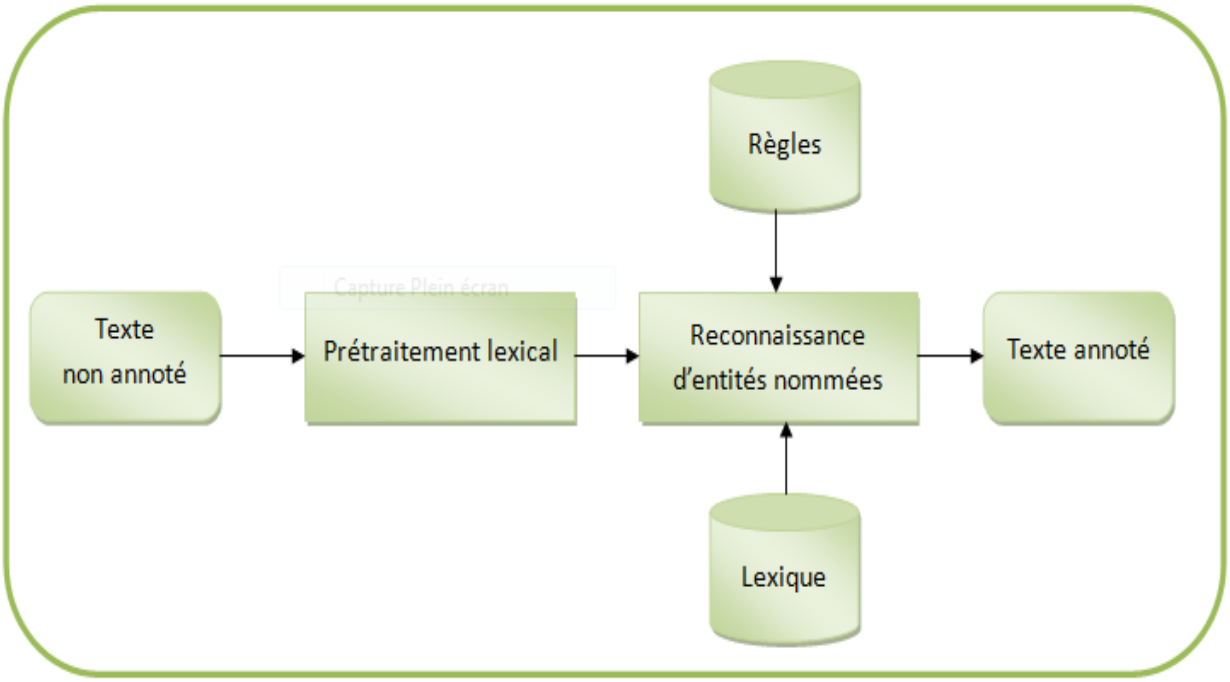


Figure2.3 : Architecture générale de Nemesis

II.5.3.2 Approches statistiques :

Ces approches sont nommées aussi approches numériques ou approches par apprentissage, ils utilisent des processus automatiques pour l'extraction d'information. Elles sont à leur tour divisées en trois types : l'apprentissage supervisé, l'apprentissage semi-supervisé et l'apprentissage non supervisé [20] :

Exemple :

On a dans le corpus d'entraînement plusieurs fois le terme abrégé « Mlle. » suivi d'un terme (ou plusieurs termes) qui est annoté comme étant une entité de type personne (EN-PERS).

Suite à cette observation, le système d'extraction des ENs va annoter les nouveaux termes précédés par le terme abrégé « Mlle.» comme des entités de type personne (EN-PERS).

II.5.3.3 Approches mixtes :

Celles-ci sont des approches hybrides qui présentent une combinaison entre les deux méthodes précédentes en faisant par exemple [16] :

- L'apprentissage de règles puis révision par un expert.
- L'élaboration de règles par un expert puis extension automatique de la couverture.

Les résultats d'un système d'extraction d'entités nommées sont généralement représentés par une structure à plat, il est ainsi possible de projeter les annotations directement sur les parties de textes concernées. Le langage XML est largement privilégié pour cette description. Il permet entre autres de garder une certaine lisibilité pour le lecteur humain grâce à l'injection de balises explicites jouant à la fois le rôle de bornes et de typage de classe [23].

Exemple :

L'homme d'affaire <Personne>**IssadRebrab**</Personne> élargit son entreprise, le groupe <Organisation>**Cevital**</Organisation> en rachetant le géant de l'électroménager <Organisation>**Brandt**</Organisation> en <Lieu>**France**</Lieu> et en augmentant sa production en <Lieu>**Algérie**</Lieu>

II.6 Représentation des entités nommées par une ontologie :

L'ontologie est une spécification explicite et formelle d'une conceptualisation qui facilite le partage de la connaissance et permettre son exploitation automatique par une machine et qui définit les termes de base, et les relations constituant le vocabulaire d'un domaine donné ainsi que les règles de combinaison de termes, et de relations pour définir des extensions du vocabulaire. Elle joue un rôle majeur dans le processus d'extraction d'information (EI). Alors que l'ontologie sémantique est une spécification basée sur la sémantique où les systèmes d'extraction d'information (EI) l'utilise pour identifier les types d'entités extraites en les reliant à leur description sémantique en se basant sur [24] :

- **Une ontologie en entrée :** le processus EI est guidé par une ontologie pour extraire des informations (catégories, relations ...) grâce à une annotation sémantique des textes à traiter.

- **Une ontologie en sortie :** le système d'EI utilise une ontologie pour représenter, stocker les informations extraites par le peuplement d'une ontologie qui consiste à ajouter de nouvelles informations (instances, propriétés, relations ...) sur les entités dans une ontologie existante.

En somme, une ontologie permet de conceptualiser de façon formelle la connaissance d'un domaine, en décrivant explicitement les concepts et les relations entre eux. Par conséquent, cette notion est très utilisée pour la représentation de différents types d'EN propres à des domaines précis. Cette tendance s'est généralisée avec l'apparition du web sémantique et le besoin de partager la connaissance sur internet.

Il existe plusieurs ontologies chacune modélise un domaine particulier d'EN comme par exemple (l'ontologie MESH représente l'EN du biomédical tandis que l'ontologie PLANTS décrit l'EN du botanique) [14].

En prenant l'exemple de la figure (2.4), l'ontologie de « **Abdelhamid Ben Badis** » se diffère sémantiquement, cette entité est parfois un nom d'une personne, une université ou une mosquée et à partir de ces différences que la recherche et l'extraction d'information est faite en réduisant l'espace de recherche et en gagnant du temps pour avoir de la pertinence en résultat.

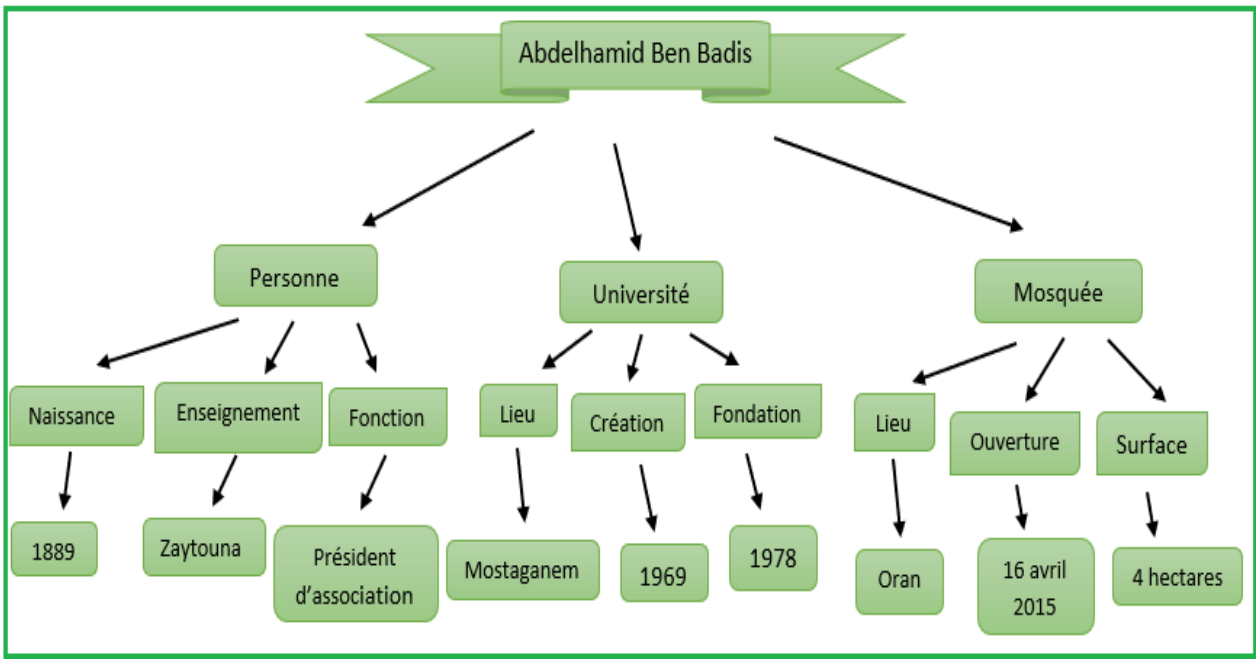


Figure 2.4 : Exemple d'ontologie sémantique

CHAPITRE 2 : ENTITEE NOMMEE

II.7 Les Système de Extraction d'EN :

II.7.1 Open Calais :

Lancé en 2008, Open Calais est un service Web gratuit fourni par Thomson Reuters. Cette boîte à outils est utilisée pour incorporer des fonctionnalités sémantiques dans des systèmes de gestion de contenu, des applications, des sites Web et même des blogs. Cette initiative identifie des individus, des sociétés, des faits et des événements [23]. Elle permet également de se connecter aux sources de données du site « Linked Data Cloud » [25].

II.7.2 Gate :

Gate est une infrastructure de développement et de déploiement de composants logiciels qui traitent le langage humain. Ce dernier est en cours de développement à l'université de Sheffield depuis 1995. Il a été utilisé dans une grande variété de projets de recherche et de développement, il a été employé dans un large éventail de contextes d'analyse linguistique, y compris l'extraction d'information dans plusieurs langues.

Gate comprend des documents, des corpus et divers types d'annotations, il prend en charge les documents dans une variété de formats, y compris XML, RTF, Email, HTML, dans tous les cas, le format est analysé et converti en un seul modèle unifié d'annotation. Enfin, il utilise des techniques à états finis pour implémenter différentes tâches allant de la tokenisation au balisage sémantique plus la comparaison des annotations.

Enfin, il a un marqueur sémantique contenant des règles qui agissent sur les annotations attribuées lors des phrases précédentes afin de produire des sorties d'entités annotées [26].

II.7.3 UIMA :

UIMA (Unstructured Information Management Architecture), est un Framework dont le but est de permettre l'analyse de données non-structurées. Il est sous licence Apache, fournissant une API sous formes de librairies jar et également des outils graphiques afin d'annoter et de visualiser les documents à annoter [27].

II.8 Variations des entités nommées :

Selon le graphisme, la syntaxe ou même le lexical, les entités nommées se varient entre elles en plusieurs variations (voir figure 2.5) :

- **Une variation graphique :**

Elle peut être aussi simple que l'utilisation ou non de majuscules (ex. : Parti Politique et Parti politique), la présence ou non de points dans les sigles ou acronymes (ex. : J.S.K et JSK) ou elle peut concerner les sigles ou les acronymes (ex. : OMS et Organisation Mondiale de la Santé).

- **Une variation syntaxique :**

Elle correspond à un changement dans l'ordre des mots (ex. : Annan Kofi et Kofi Annan) ou à l'alternance entre des paires syntaxico-sémantiquement équivalentes (ex. : le roi de l'Espagne et le roi Espagnol).

- **Une variation lexicale :**

C'est une normalisation plus complexe où ces variations sont amalgamées avec d'autres variations comme par exemple (Recep Tayyip Erdogan, R. Tayyip Erdogan, R.T. Erdogan et président de la république turque).

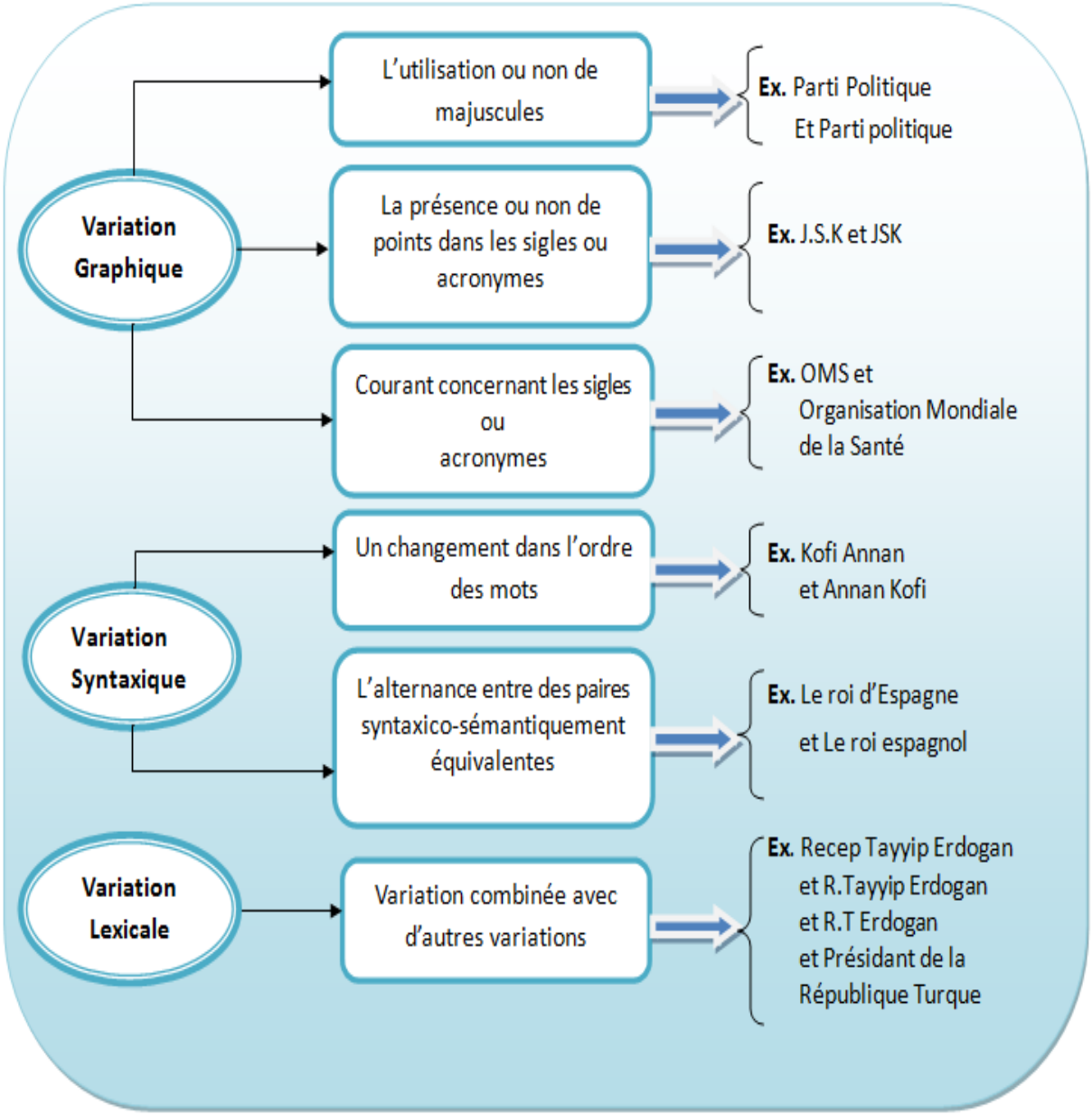


Figure 2.5: Variations des Entités nommées.

CHAPITRE 2 : ENTITEE NOMMEE

II.9 Les limites et les problématiques des EN (Ambiguïté) :

Parmi les problèmes d'entités nommées qui complexifient l'annotation des entités nommées est l'ambiguïté où l'entité nommée peut avoir plusieurs interprétations en délimitation ou typologie, ils existent différents problèmes d'ambigüité parmi eux on cite [22] :

II.9.1 Ambiguïtés graphiques :

Elles concernent le graphisme et le format de l'entité nommée comme par exemple la majuscule qui est considéré comme un indicateur pour le repérage et la délimitation des ENs et qui n'est pas simple à manipuler pour plusieurs raisons tel que :

- Une entité nommée peut contenir des formes commençant par une minuscule par exemple « les jardins de Babylone ».
- Une EN peut comporter une ou plusieurs majuscules dans d'autres positions par exemple « LaTeX ».
- La première forme d'une phrase comporte aussi une majuscule, que ce soit une entité nommée ou non.
- L'emploi de la majuscule pour les noms propres n'est pas de règle dans toutes les langues.

II.9.2 Ambiguïtés sémantiques :

A l'exemple des noms communs, prenons les exemples suivants :

- Orange a été inauguré en 1988.
- Les œuvres de Dib ont marqué la littérature algérienne.
- L'Algérie a signé le traité d'Evian.
- DSP (Direction de la Santé et de la Population).

Ces phénomènes complexifient la tâche de catégorisation, par exemple **Orange** s'agit-il de l'entreprise de télécommunication ou bien de l'école spécialiste en télé-opération ? De la personne de **Mohammed Dib** ou de l'Université **Mohammed Dib**? Est-ce que l'**Algérie** est traité comme un « lieu » ou bien un « gouvernement », de même pour **Evian** « un lieu » ou « un traité » ?

II.10 Conclusion :

Dans ce chapitre nous avons défini les unités appelées entités nommées (ENs), nous avons présenté la reconnaissance de ces entités ainsi que l'extraction d'information qui est une étape importante pour cette dernière. Nous allons par la suite clôturer le rapport avec une conclusion générale.

CHAPITRE 3 :
CONCEPTION

CHAPITRE 3 : CONCEPTION

III.1. Introduction :

L'objectif de ce travail est de développer un système de recherche d'information, capable de retourner des documents numériques pertinents pour un utilisateur exécutant une requête en se basant sur des entités nommées ou des mots clés de la requête. Pour cela, nous avons choisi d'utiliser le logiciel GATE.

Dans ce chapitre, nous avons organisé notre travail en deux parties : la première partie consiste à définir le logiciel GATE que nous avons utilisé pour faire le traitement des documents. Nous présentons ses composants, ses applications et ses fonctionnalités. La deuxième partie comprend notre modélisation et notre contribution du système de recherche d'information par entité nommées en utilisant GATE.

III.2. Définition de GATE :

Sous le nom de GATE, General Architecture for Text Engineering, est regroupée en une suite d'outils développés en java à l'université de Sheffield en 1995. Ces outils sont utilisés pour toutes sortes de tâches de traitement de langage naturel, y compris l'extraction d'information dans de nombreuses langues. GATE aide les scientifiques et les ingénieurs à définir et à déterminer les structures organisationnelles du traitement du langage et facilite l'intégration des capacités de traitement du langage dans les applications [29].

Comme tous les logiciels, GATE a une interface graphique (GUI) qui aide les utilisateurs a bien travaillé avec lui, voici son interface éclaircie dans la figure suivante :

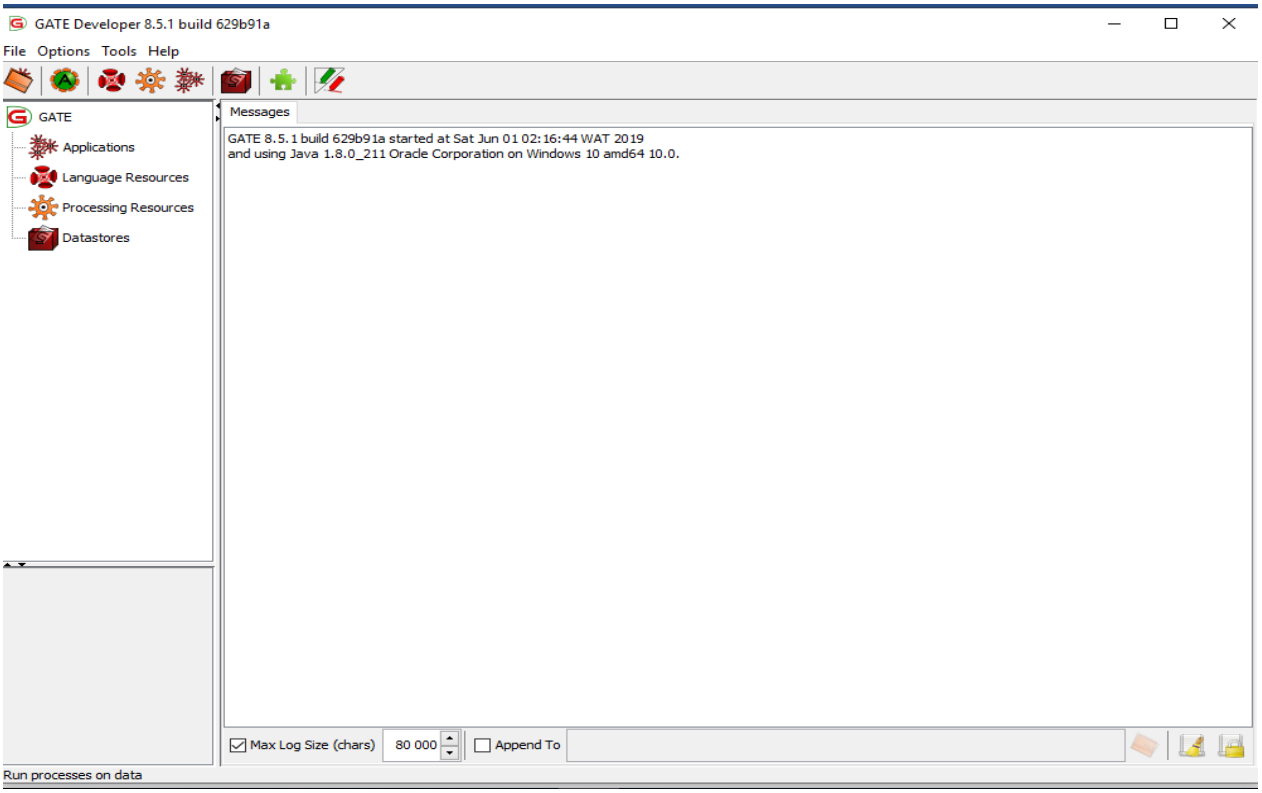


Figure 3.1 : Interface de GATE

III.3. Les composants de GATE :

GATE comprend des composants pour diverses tâches de traitement de langues, telles que les analyseurs syntaxiques, la morphologie, le balisage, des outils de récupération d'informations, des composants d'extraction d'informations pour différentes langues, la figure suivante illustre ces derniers [29] :

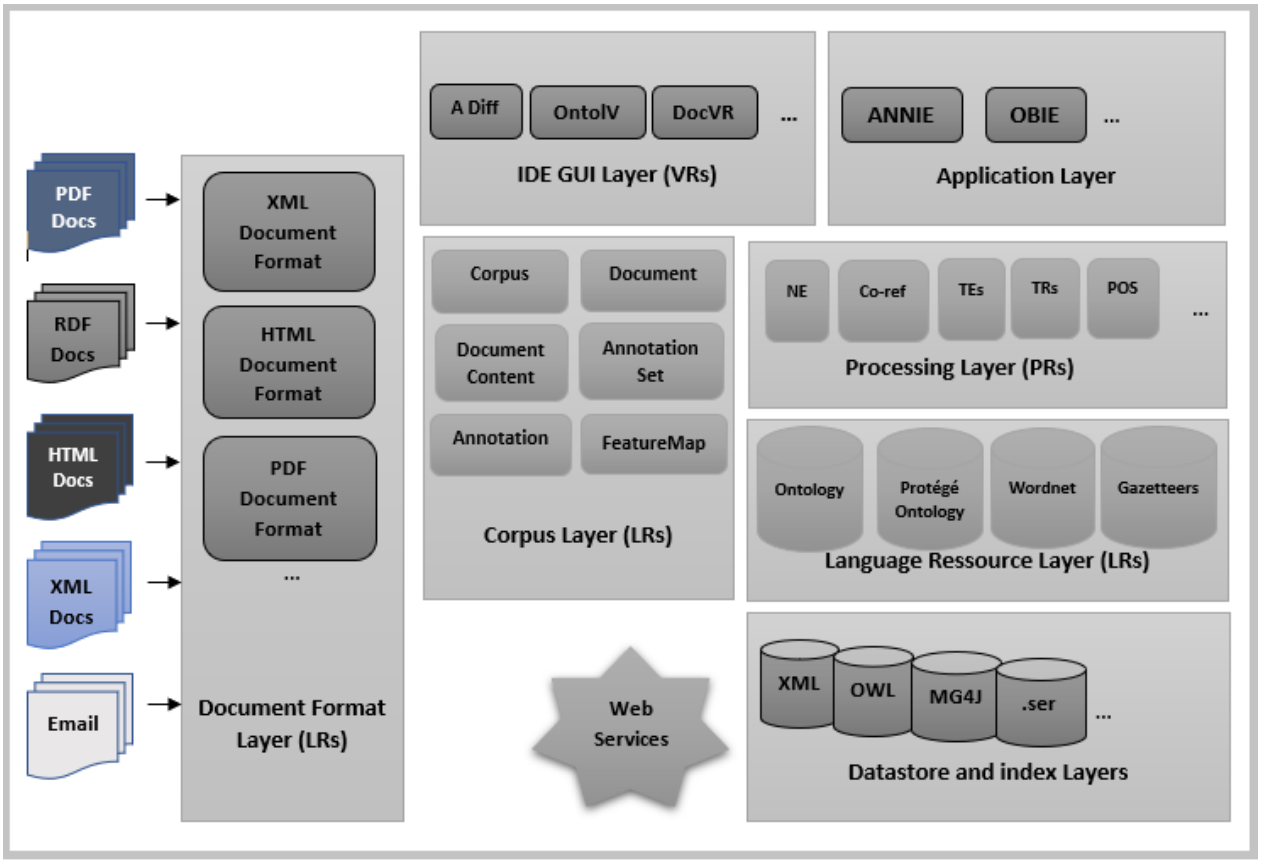


Figure 3.2 : Les composants de GATE

III.3.1. Document Format Layer (LRs) :

GATE a la possibilité d'importer du texte qui est l'une des fonctionnalités les plus importantes des logiciels d'analyse de texte, car les utilisateurs doivent extraire des données textuelles de différentes sources. Ce logiciel d'exploration de données peut contenir des données importantes dans différents formats tels que texte brut, HTML, PDF, RTF, XML.

III.3.2. IDE GUI Layer (VRs):

III.3.2.1. Annotation Diff (A Diff):

Outil de l'environnement de développement qui implémente des métriques de performance telles que la précision et le rappel pour la comparaison d'annotation. Généralement, un développeur

d'analyse linguistique marque certains documents à la main, qu'il utilise ensuite avec l'outil « A Diff » pour mesurer automatiquement les performances des composants.

III.3.3. Application Layer :

Les applications de GATE sont exécutées sur des documents ou des corpus de documents importés dans GATE, elle exécute certains types de ressources de traitement de texte. Il existe plusieurs types d'applications, certaines sont intégrées déjà dans GATE, d'autres doivent être importées, ces dernières ont leurs propres ressources de traitements, et toutes ces applications peuvent être utilisées pour le traitement automatique des textes.

Les applications existantes dans GATE sont des applications pipeline où un pipeline est constitué d'une chaîne d'éléments de traitement (processus, threads, fonctions…etc.) organisés séquentiellement de manière à ce que la sortie de chaque élément soit l'entrée du suivant, ces pipelines sont de trois types (voir la figure 3.2) :

- **Pipeline :**

Une application de pipeline ne peut être exécutée que sur un seul document, alors qu'un pipeline de corpus peut être exécuté sur un corpus entier.

- **Conditional pipeline :**

Sont des versions conditionnelles les pipelines précédentes et permettent d'exécuter ou non les ressources de traitement en fonction de la valeur d'une caractéristique du document.

- **Real-Time pipeline :**

Où le temps réel est utile lorsque vous traitez des données à partir d'une source de diffusion en continu, telles que les données de marchés financiers ou la télémétrie à partir de périphériques connectés.

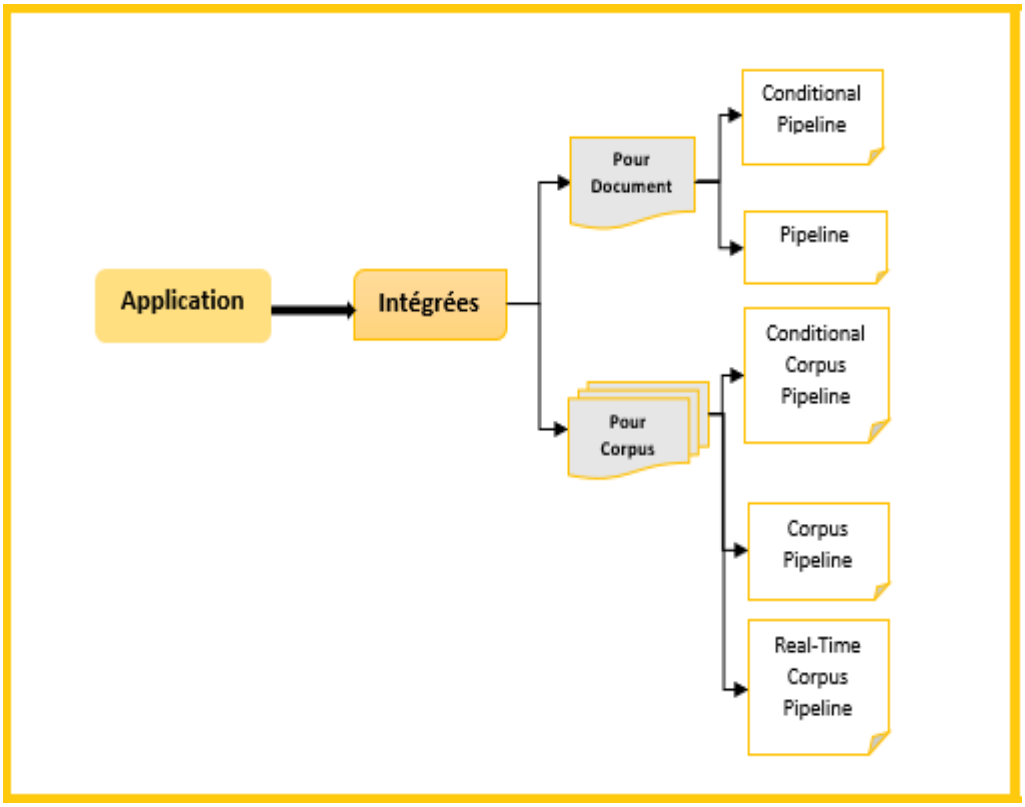


Figure 3.3 : Les types des applications de GATE

CHAPITRE 3 : CONCEPTION

Parmi les applications les plus utilisées par GATE, nous citons ANNIE (A Nearly-New Information Extraction System) qui est un système presque nouveau d'extraction d'information (IE), il était basé sur un système d'IE déjà existant sous le nom de « LaSIE », ce système est une collection de ressources de traitements (PRs) prête à l'emploi qui s'exécutent sur des documents (txt, xml, html …), ces ressources sont illustrées dans la figure suivante :

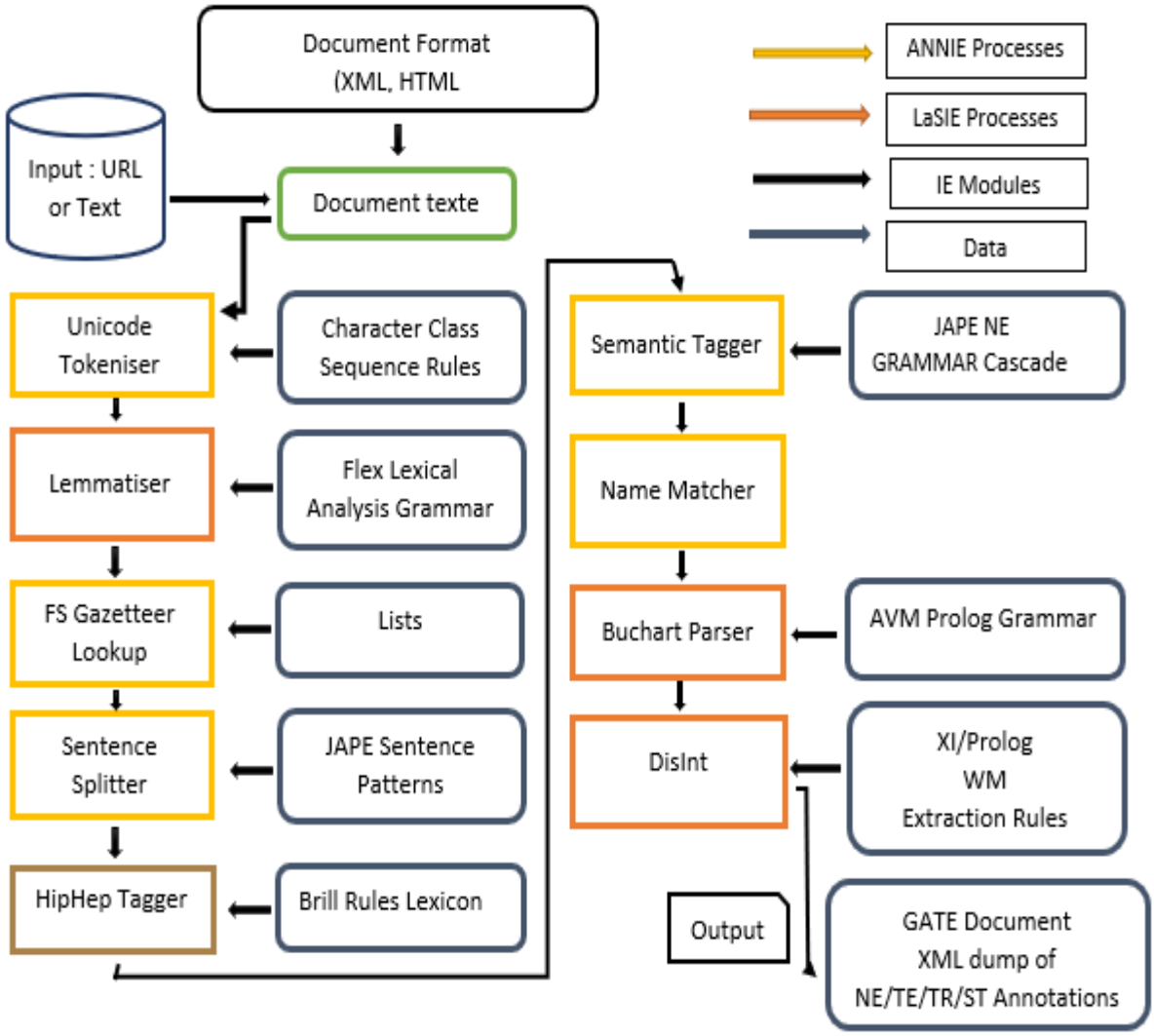


Figure 3.4 : Les composants de l'application ANNIE

Comme il est défini dans la figure ci-dessus, ANNIE se compose d'un Tokeniser, d'un séparateur de phrase « Sentence Splitter », d'un étiqueteur POS « POS Tagger », d'un répertoire géographique « Gazetteer », d'un transducteur à états finis « Finite State Transducer », d'un orthomateur « Orthomatcher » et d'un Coreferencer où le Tokeniser divise le texte en tokens c'est-à-dire des nombres, des signes de ponctuation, des symboles et des mots de types différents, le Sentence Splitter segmente le texte en phrases, qui constituent l'entrée du tagueur, Le marqueur sémantique se compose de règles artisanales décrivant les modèles à mettre en correspondance et

les annotations à créer en conséquence. L'Orthomatcher effectue une Co-référence, ou suivi d'entité, en reconnaissant les relations entre les entités et Le coreferencer trouve des relations d'identité entre des entités dans le texte.

Ces ressources de traitements se chargent automatiquement lors du chargement de l'application ANNIE (voir figure 3.5) :

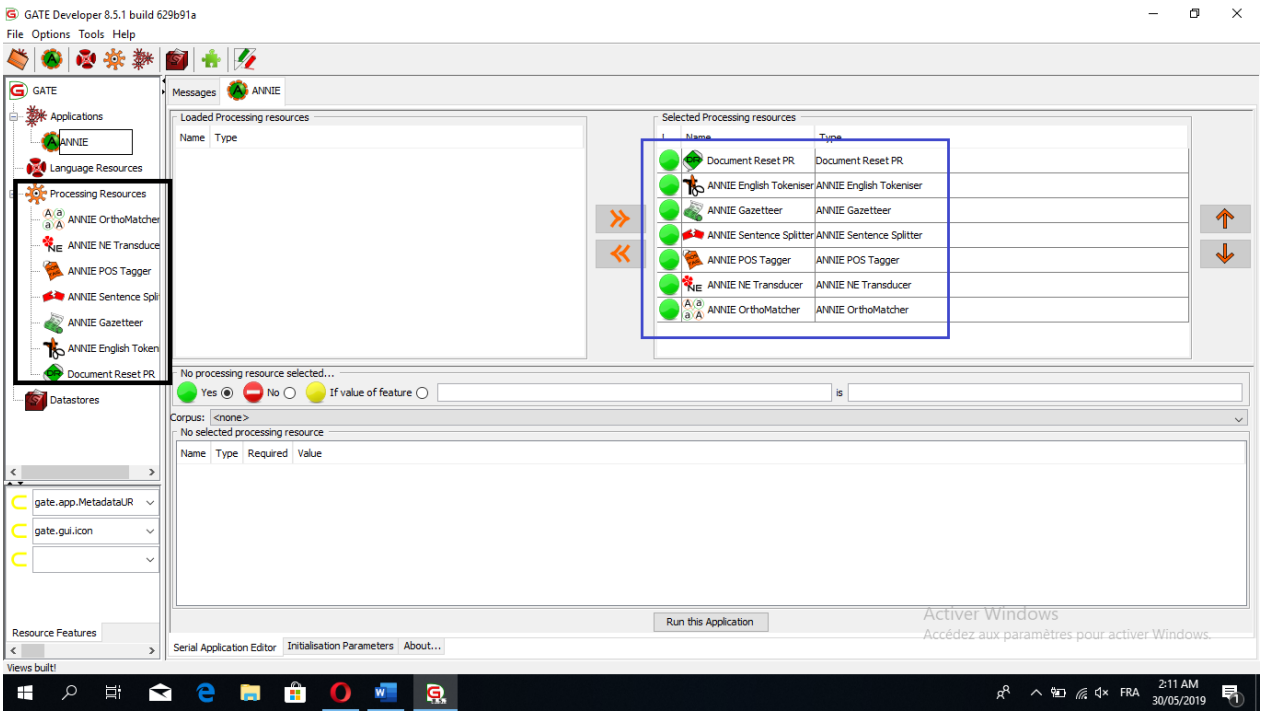


Figure 3.5 : Processing Ressources de l'application ANNIE

GATE fournit des fonctionnalités faciles à utiliser et extensibles pour l'annotation de texte afin d'annoter les données d'apprentissage requises pour les algorithmes de NLP (Natural Language Processing). L'annotation peut être faite manuellement par l'utilisateur ou semi-automatiquement en exécutant certaines ressources de traitement sur le corpus, puis en corrigeant et en ajoutant de nouvelles annotations manuellement. Selon les informations à annoter, certains modules ANNIE peuvent être utilisés ou adaptés pour amorcer la tâche d'annotation de corpus.

GATE aborde l'ensemble des problèmes liés au développement d'applications NLP de manière flexible et extensible. Il favorise la robustesse, la réutilisation et l'évolutivité.

III.3.4 Processing Layer (PRs) :

Les ressources de traitement (Processing Resources PRs) représentent des entités principalement algorithmiques, telles que des analyseurs syntaxiques, des générateurs ou des modélisateurs ngram. Ils sont créés à l'aide de GATE Factory de manière similaire aux ressources linguistiques. Outre les paramètres de création, ils disposent également d'un ensemble de paramètres d'exécution définis par le système juste avant de les exécuter. Les analyseurs sont un

type particulier de Processing Resources en ce sens qu'ils ont toujours un document et un corpus parmi leurs paramètres d'exécution.

Ces ressources sont fournies généralement par les applications ajoutées ou appelés sur GATE, comme nous avons montré dans l'exemple de la figure 3.4.

III.3.5 Language Ressources (LRs) :

Désigne des ressources contenant uniquement des données, telles que des lexiques, des corpus, des thésaurus ou des ontologies. Certains LR sont livrées avec un logiciel (par exemple Wordnet possède à la fois une interface de requête utilisateur et des APIs C et Prolog), mais lorsqu'il ne s'agit que d'un moyen d'accéder aux données sous-jacentes, ces ressources sont toujours définies comme étant des LRs.

III.3.6 Corpus Layer :

Elle comprenne les documents et les processus de traitement exécutés sur ces documents où :

- **Documents :** des fichiers de différents types (txt, xml, pdf...) à annoter, voilà un exemple :

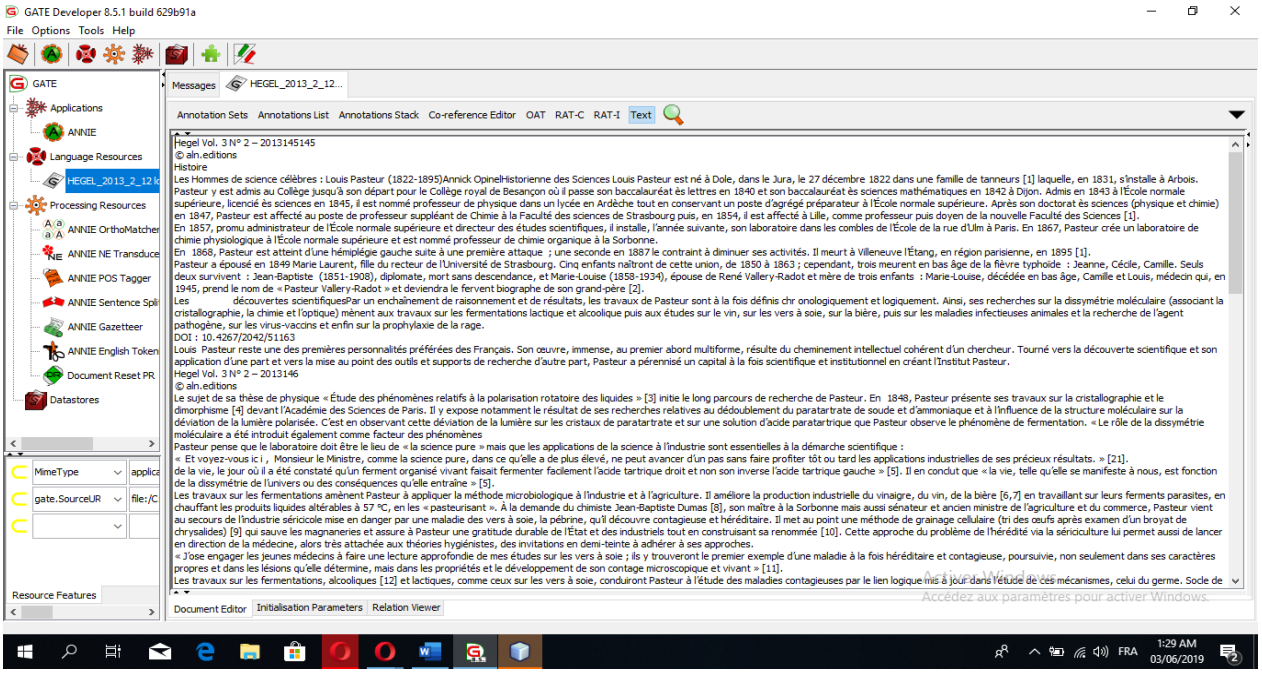


Figure 3.6 : Exemple de document importé dans GATE

- **Corpus de documents :** un ensemble de documents à annoter qui peuvent être sur le même thème ou non, comme il est défini sur la figure suivante :

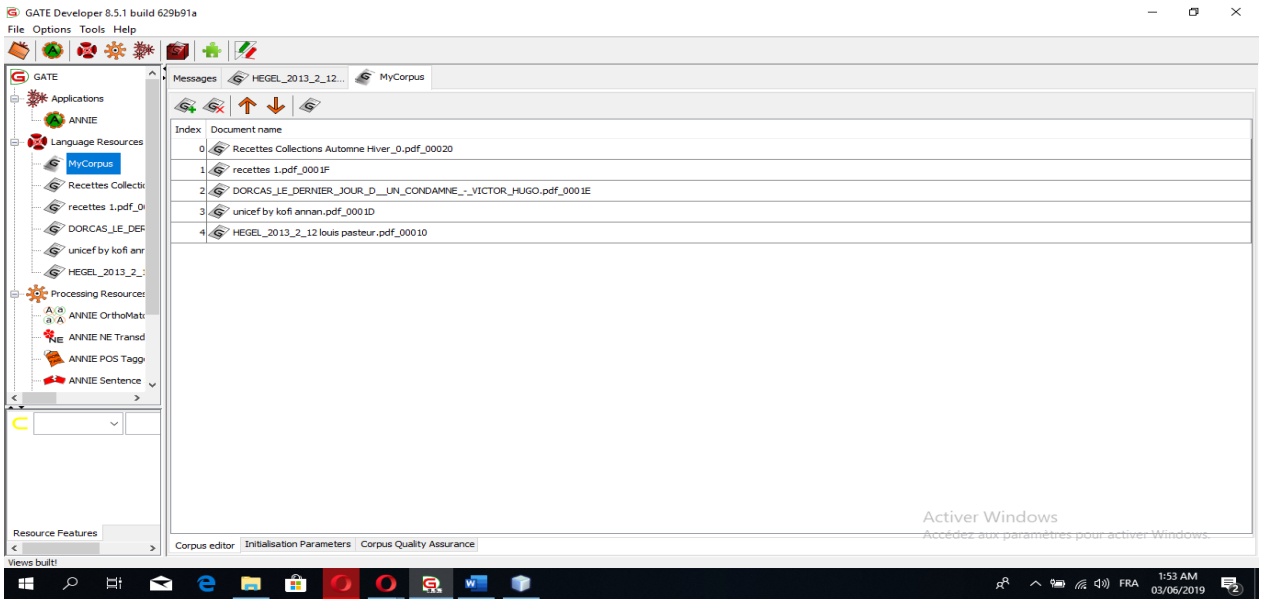


Figure 3.7: Exemple d'une liste de documents d'un corpus sur GATE.

- **Document Content :** le contenu des documents à annoter, celui qui apparaît pour le traitement sur Document Editor (voir la figure 3.5)).
- **Annotation Set :** les groupes d'annotations exécutées sur les documents, voir la figure suivante :

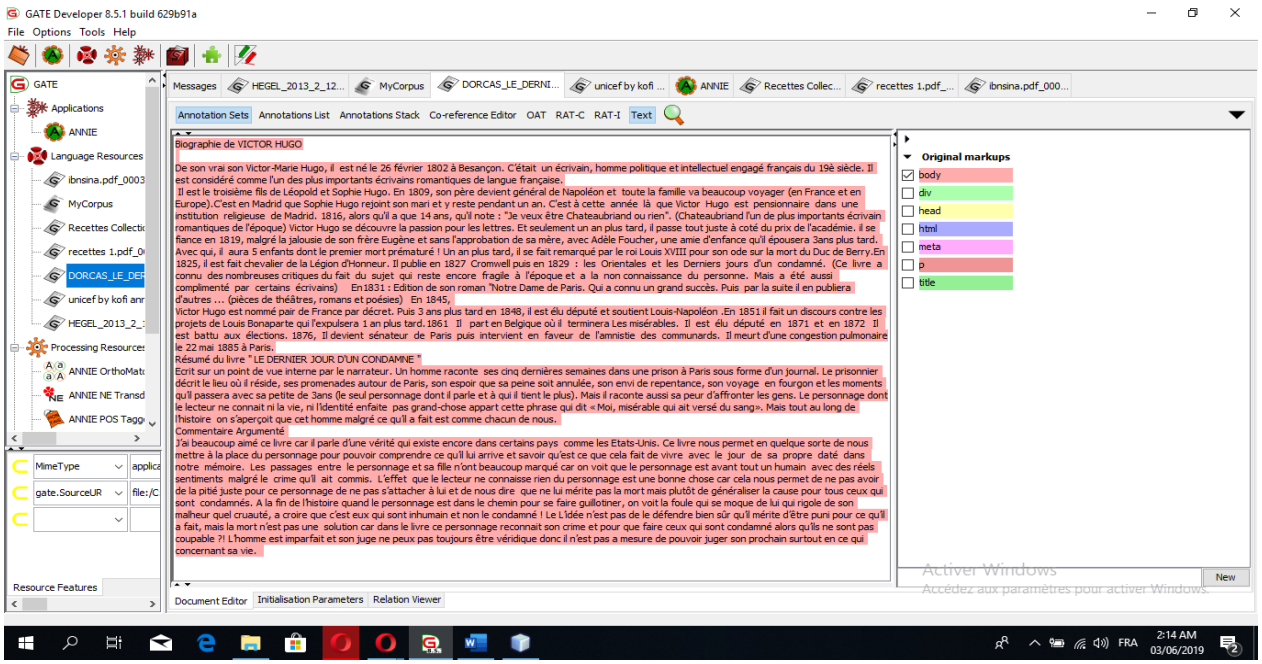


Figure 3.8 : Exemple d'Annotations Set

- **Annotation :** Les annotations sont des commentaires, des notes, des explications ou d'autres types de remarques externes pouvant être attachés à un document numérique ou à

une partie sélectionnée d'un document. D'un point de vue technique, les annotations sont généralement considérées comme des métadonnées, car elles fournissent des informations supplémentaires sur une donnée existante.

- **Feature Map :** Toutes les ressources de traitement, ainsi que les contrôleurs et les annotations, peuvent être associées à des métadonnées sous forme de cartes de caractéristiques. Une Feature Map (carte de fonction) est une carte Java qui contient des paires <nom-attribut, valeur-attribut>.

III.3.7 Datastore and Index Layer:

Les datastores servent à stocker des documents et des corpus sur le disque dur, chaque datastore correspond à un répertoire sur le disque. Les fichiers stockés dans le répertoire sont au format GATE. Il existe trois types de datastores dont chacune est spécifiée par rapport aux autres :SerialDataStore, Searchabledatastore et LuceneBased qui enregistre les documents traités en index, dans la figure suivante un exemple de la base de type Searchabledatastore :

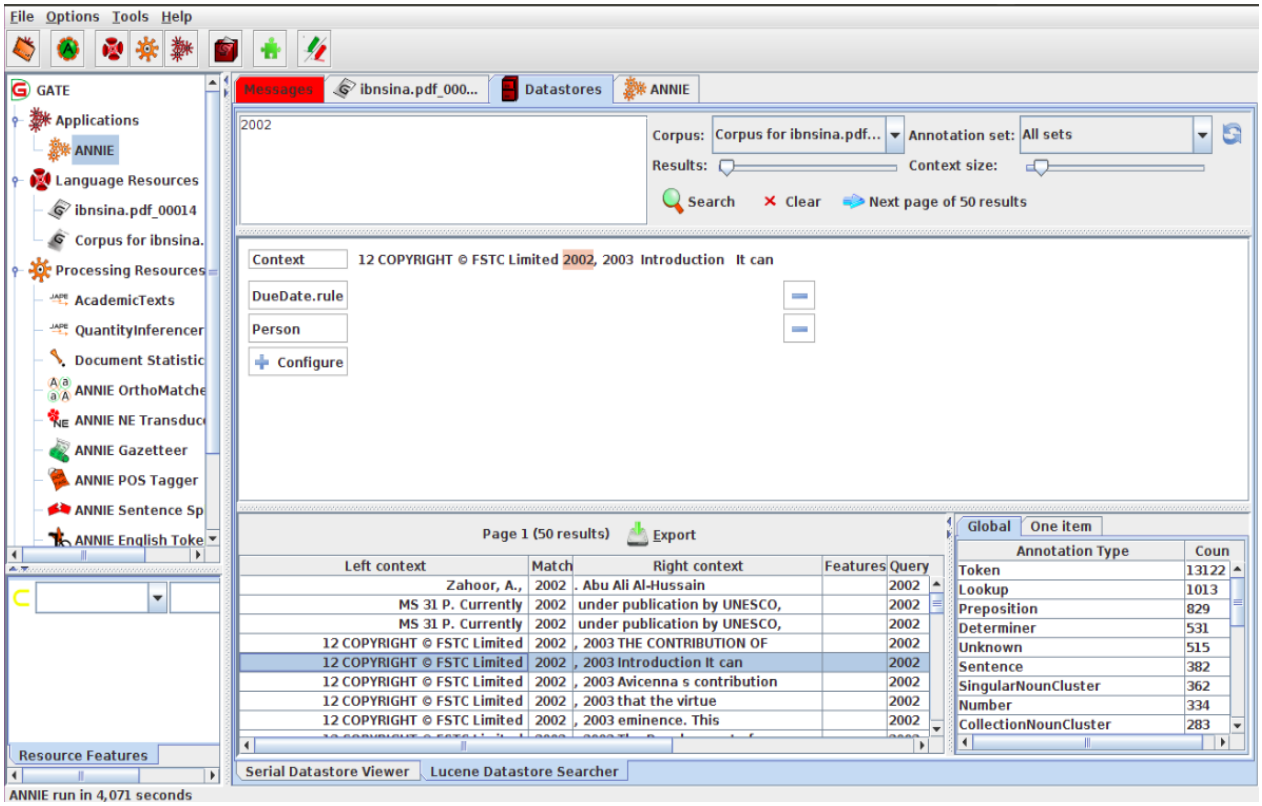


Figure 3.9: un exemple de datastore

L'exécution d'une application sur un corpus stocké dans une datastore enregistre automatiquement chaque document traité.

III.3.8 Web Service :

La technologie des services web est un moyen rapide de distribution de l'information entre clients, fournisseurs, partenaires commerciaux et leurs différentes plateformes, basée sur le modèle SOA. Ces services fournissent un lien entre applications. Ainsi, les applications utilisant des technologies différentes peuvent envoyer et recevoir de données au travers des protocoles compréhensibles par tout le monde. Parmi ces protocoles, Gate utilise le protocole SOAP (Simple Object Access Protocol), qui est un protocole standard de communication, décrit en XML et standardisé par le W3C. Il se représente comme une enveloppe pouvant contenir des données et qui se circule sur le protocole http, permettant d'effectuer des appels de méthodes à distance.

GATE offre des outils très variés afin de traiter le plus de problème de linguistique possible, en allant de la simple annotation de texte au travail sur les ontologies, la figure suivante les explique plus précisément [29] :

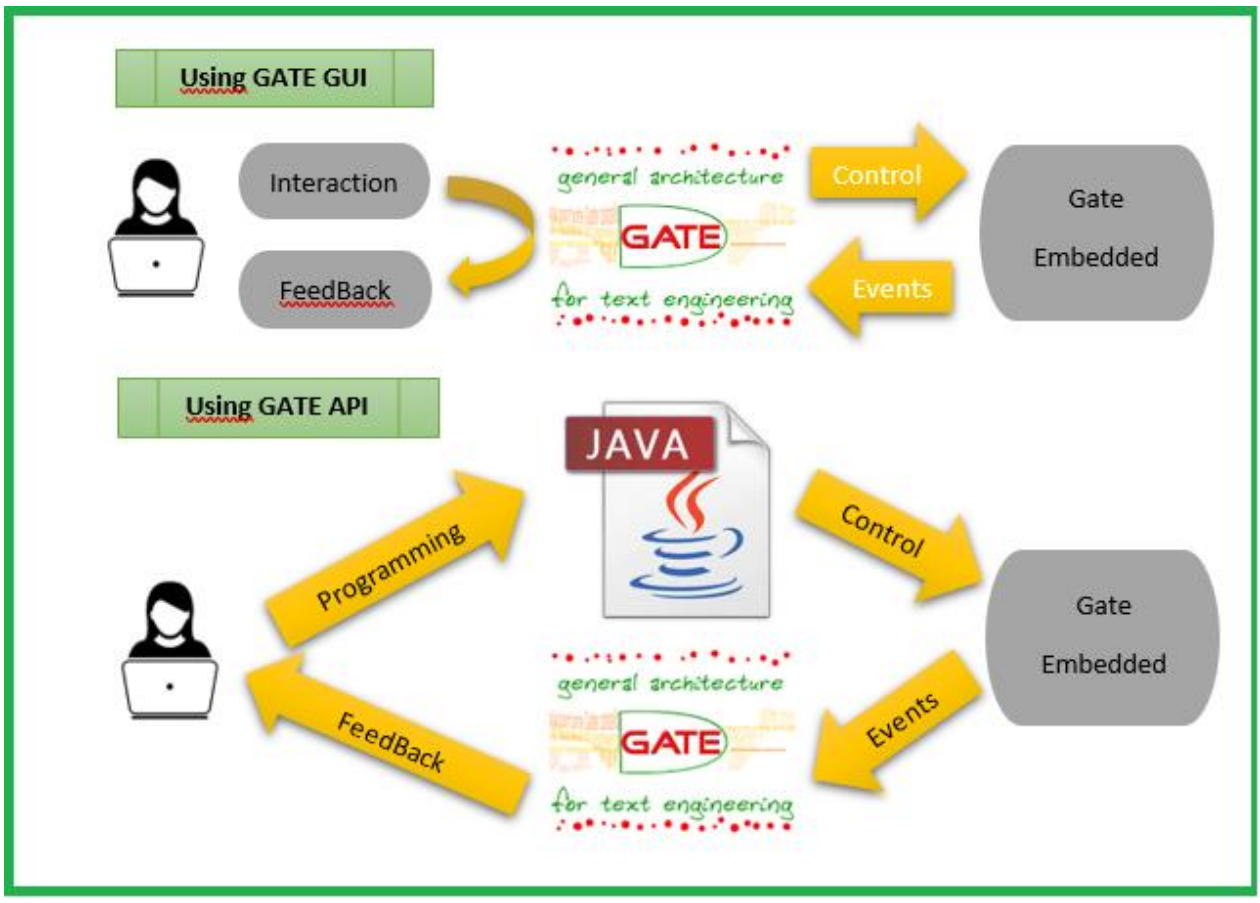


Figure 3.10 : Les outils de GATE

Nous pouvons notamment citer GATE Developers, qui est un environnement de développement avec une interface graphique qui sert principalement à annoter des documents (extraction d'information).

Nous citerons GATE Embedded qui est la librairie permettant d'utiliser tous ces outils dans une application, c'est-à-dire un Framework (ou une bibliothèque de classes) orienté objet implémenté en Java. Il est utilisé dans tous les systèmes basés sur GATE et constitue l'élément de base (non visuel) de GATE Developers.

Comme son nom l'indique GATE Embedded est conçu pour vous permettre d'intégrer des fonctionnalités de traitement de langage dans diverses applications. C'est un outil de programmation fourni sous la forme d'un ensemble d'archives Java (JAR).

Ensuite, nous définissons notre démarche utilisée dans ce développement en citant quelques problématiques qu'on a rencontré en utilisant GATE :

III.4 Notre démarche :

Etant donné que GATE a plusieurs outils qu'on peut utiliser pour traiter les textes, extraire des informations, et même pour implémenter un système de recherche d'information. Nous avons utilisé ce logiciel pour développer notre système de recherche d'informations basé sur les entités nommées et les mots clés.

Dans ce travail, nous sommes passés par plusieurs étapes qui sont définis dans la figure suivante :

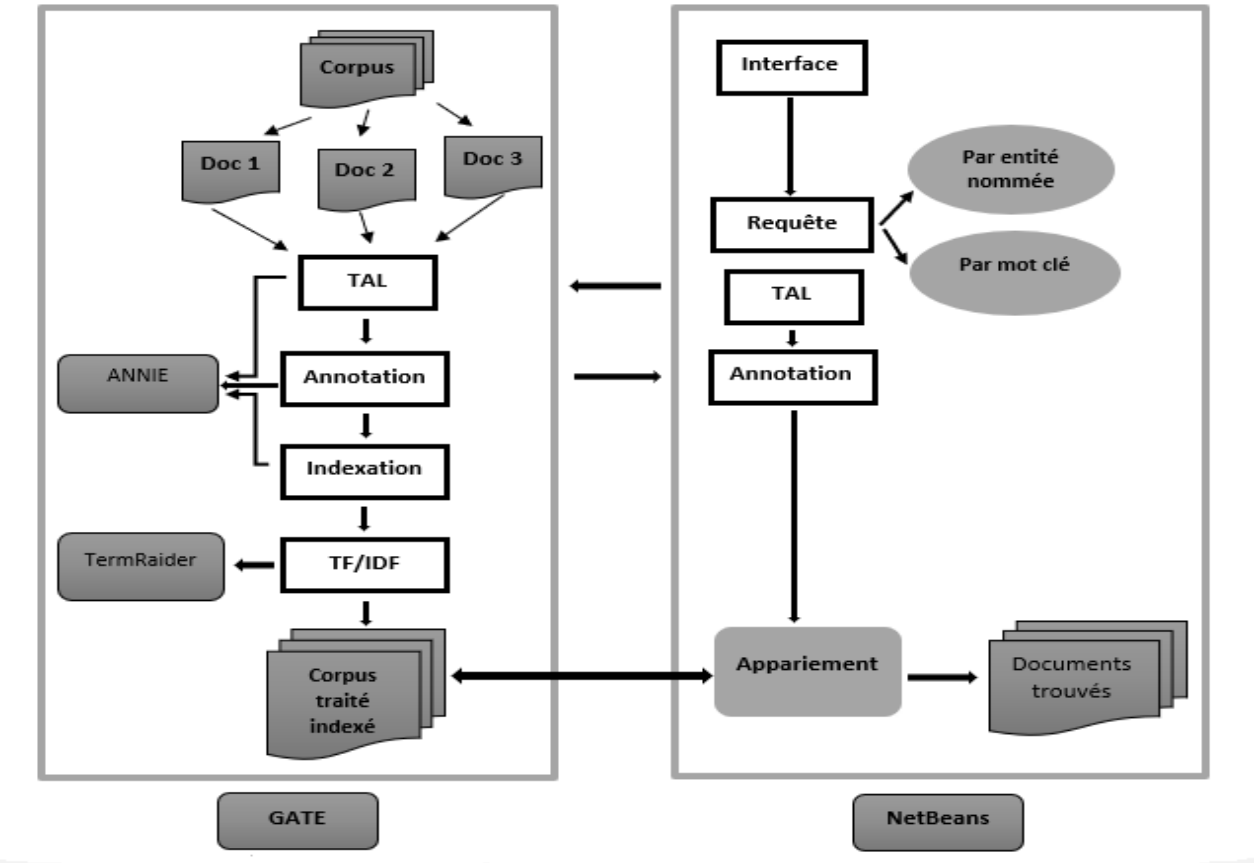


Figure 3.11: Notre démarche

CHAPITRE 3 : CONCEPTION

III.5 Explication de notre démarche :

Comme la figure le montre, nous sommes passés par plusieurs étapes durant le processus de développement de notre système, ces derniers sont :

III.5.1 Traitement de la base du système :

Cette phase est faite avec le logiciel GATE :

a. **Corpus :**
   Premièrement, nous avons téléchargé un ensemble de documents sur internet, ces documents étaient de différents thèmes et extensions (pdf, txt, xml…).
b. **Traitement des corpus :**
   Ensuite, nous avons importés les documents téléchargés dans le logiciel GATE pour les traiter et les indexer. Alors, pour ce traitement nous avons chargé des applications dans GATE (ANNIE…) puis nous les avons exécutés. Ces applications traitent le langage des documents automatiquement (stopwords, tokens…), et annotent le texte de ces documents automatiquement et manuellement (organization, person, location, date…).
c. **Indexation :**
   Nous avons sauvegardé les documents traités avec ses traitements dans des base de données (datastores) fournies par GATE lui-même. Ces datastores indexent automatiquement les documents traités.
d. **TF/IDF :**
   Et enfin, nous avons classifié les documents par pertinence en utilisant le TF/IDF.
   Le TF/IDF a un processus que nous le définissons dans cet algorithme :

Algorithme du TF/IDF :

Données :

Documents, Corpus, …

Début :

Tant que :

New document ;

Fait :

Traiter document (TAL) ;

Annoter document ;

Calculer nombre de mots du document ;

Calculer nombre d'occurrence de chaque entité nommée dans le document ;

Calculer fréquence d'apparition de chaque EN ;

Comparer les documents par rapport à la fréquence d'apparition calculée ;

Classer les documents par rapport à cette fréquence ;

Fin.

III.5.2 Implémentation du système :

Pour implémenter notre système, nous avons utilisé le logiciel NetBeans avec le langage Java :

a. **Interface :**
   Nous avons implémenté une interface simple composée d'un TextField pour que l'utilisateur puisse taper sa requête pour effectuer une recherche, en plus il a la possibilité de choisir entre deux types de recherche (recherche par entité nommée ou recherche par mot clé).
b. **Traitement de la requête :**
   Nous avons importé le package « GATE » sur NetBeans pour effectuer le même traitement de documents sur la requête.
c. **Appariement :**
   Notre système va faire l'appariement des mots clés et des entités nommées de la requête avec celles des documents, pour afficher les documents équivalents à la requête de l'utilisateur, ces documents sont classés par pertinence du thème recherché.

CHAPITRE 3 : CONCEPTION

Nous présentons un algorithme qui explique et résume notre démarche dans ce travail :

Algorithme du système de RI

Données :

Documents, Corpus, …

Début :

Tant que :

Entrer new document ;

Fait :

Charger document dans GATE ;

Charger document dans Corpus ;

Charger les applications de traitement ;

Charger Processing Ressources ;

Exécuter les applications de traitement sur le corpus ;

Sauvegarder sur les datastores ;

Fin tant que.

Tant que :

Entrer Requête dans l'interface ;

Fait :

Traiter requête ;

Appariement (Documents/ Requête) ;

Fin tant que.

Documents trouvés.

Fin.

III.6 Problèmes rencontrés avec GATE :

Comme tous les outils et les logiciels, GATE n'est pas un logiciel complet. D'après l'étude que nous avons effectué sur ce dernier, nous avons rencontré plusieurs difficultés parmi eux :

- L'annotation dans Gate ne se fait pas toujours de manière correcte, surtout pour les documents qui ne sont pas présentés en anglais (par exemple il n'annote pas les mois de l'année en français en tant que date…).
- Le logiciel considère une adresse physique ainsi qu'une adresse mail comme étant une entité de type adresse URL.
- Les applications existantes dans ce logiciel ne peuvent pas calculer le TF/IDF.
- Lors de l'importation du document dans le logiciel, la plupart du temps, le texte ne s'affiche pas comme il est dans le « Document Editor », GATE sépare les phrases et même les mots du texte et fait des sauts de lignes aléatoirement.
- Parfois, il annote les entités de type tokens comme étant des entités de type personne.

III.7 Nos solutions :

D'après les recherches que nous avons effectuées, nous avons pensé à ces solutions :

- GATE a la possibilité d'ajouter des plugins et des applications pouvant faire une annotation plus au moins correcte par rapport aux applications déjà existantes (bien qu'elle n'existe pas une annotation 100% correcte). Nous avons dans notre cas ajouté une application avec une extension «.xgapp » sous le même nom de celle fournie par GATE « ANNIE ».
- Nous avons ajouté l'application « TermRaider » qui fait le calcul du TF/IDF pour faire le classement par pertinence des documents.

III.8 Conclusion :

Nous avons expliqué dans ce chapitre notre modélisation du développement de notre système de recherche d'information, nous avons décrit le logiciel GATE que nous avons utilisé pour le traitement de notre base de documents ainsi que notre processus suivi pour implémenter ce système, nous montrons dans le chapitre suivant un exemple illustré de notre système en passant par notre processus utilisé.

CHAPITRE 4 :
IMPLEMENTATION

CHAPITRE 4 : IMPLEMENTATION

IV.1 Introduction :

Ce chapitre s'intéresse aux différentes étapes dont nous sommes passées durant l'implémentation de notre système de recherche d'information, nous décrirons notre environnement de travail, ensuite nous expliquons en exemple notre allure du développement du système en le détaillant avec des interfaces.

IV.2 Environnement de travail :

L'implémentation de notre application a été réalisé dans les environnements suivants :

IV.2.1 Environnement matériel :

Nous avons utilisé deux laptops qui ont les spécifications suivantes :

- **Premier laptop :**
  - ✓ PC Dell
  - ✓ Processeur Intel® Core™ i3-4005U CPU @ 1.70GHz 1.70GHz
  - ✓ Mémoire RAM 4,00 Go
  - ✓ Windows 10 (64 bits)
- **Deuxième laptop :**
  - ✓ PC Dell
  - ✓ Processeur Intel® Core™ i3-5005U CPU @2.00 GHz x 4
  - ✓ Mémoire 3,9 Go
  - ✓ Ubuntu 16.04 (64 bits)

IV.2.2 Environnement logiciel :

En Plus de l'utilisation du logiciel GATE, nous avons utilisé le langage java sous l'environnement NetBeans, puisqu'il permet de développer des interfaces graphiques et il offre aux développeurs un environnement orienté objet visuel et événementiel :

- **Langage Java :**

  Nous avons choisi d'utiliser le langage Java car c'est un outil indispensable qui permet aux développeurs [30] :

  - ✓ D'écrire des logiciels sur une plate-forme et de les exécuter sur pratiquement toutes les autres plates-formes,
  - ✓ De créer des programmes qui peuvent être exécutés dans un navigateur Web et accéder aux services Web disponibles,
  - ✓ De développer des applications côté serveur pour des forums, des magasins et des sondages en ligne, pour le traitement de formulaires HTML, etc.,
  - ✓ De combiner des applications ou des services basés sur le langage Java pour créer des applications ou des services très personnalisés,
  - ✓ D'écrire des applications puissantes et efficaces pour les téléphones portables, les processeurs à distance, les microcontrôleurs, les modules sans fil, les capteurs, les passerelles, les produits de consommation et tous les autres types de dispositif électronique.

CHAPITRE 4 : IMPLEMENTATION

- **Environnement NetBeans :**

L'EDI NetBeans est un environnement de développement - un outil pour les programmeurs pour écrire, compiler, déboguer et déployer des programmes. Il est écrit en Java - mais peut supporter n'importe quel langage de programmation. Il y a également un grand nombre de modules pour étendre l'EDI NetBeans. L'EDI NetBeans est un produit gratuit, sans aucune restriction quant à son usage [31].

IV.3 Nos exemples illustrés :

Nous sommes passés par plusieurs étapes durant le développement de notre système, qui sont présenté ci-dessous :

- Nous avons téléchargé un corpus des documents de différents thèmes et types, nous montrons un exemple dans la figure suivante :

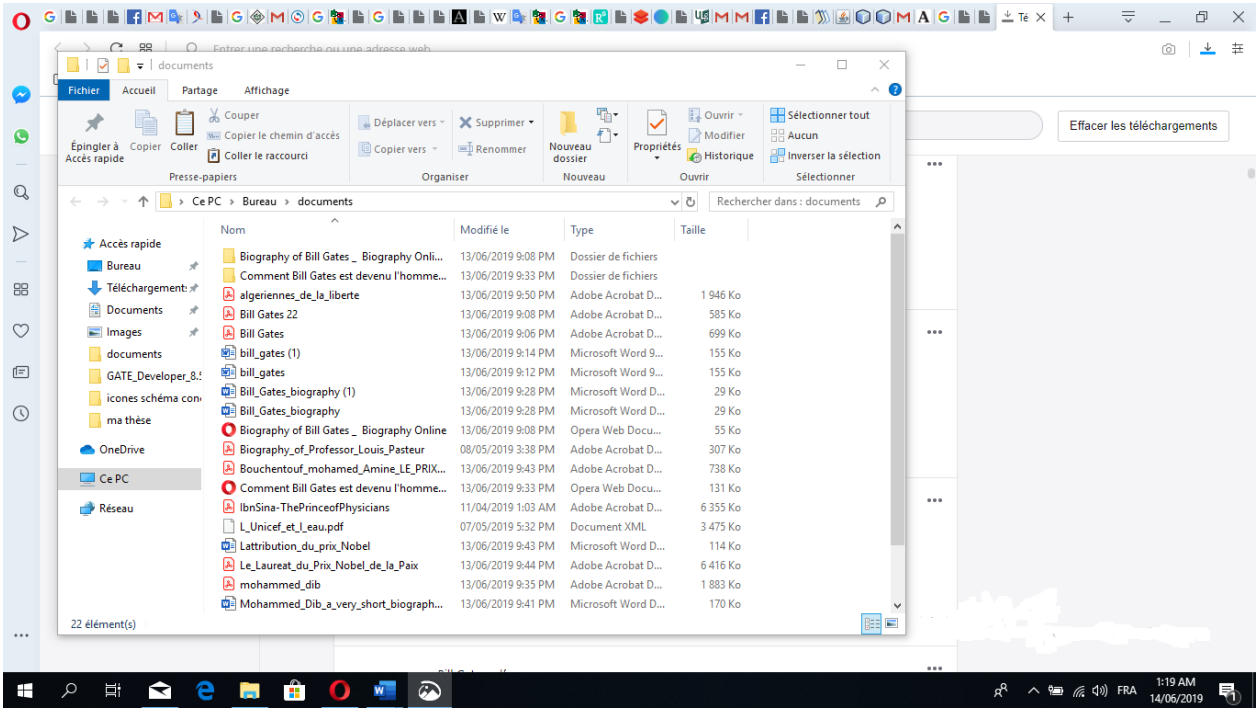


Figure 4.1 : exemple des documents téléchargés

CHAPITRE 4 : IMPLEMENTATION

Ensuite, nous avons importé les documents dans des corpus sur GATE, et nous avons les annotés :

- Gate permet de reconnaitre dans son annotation les entités de type Personne :

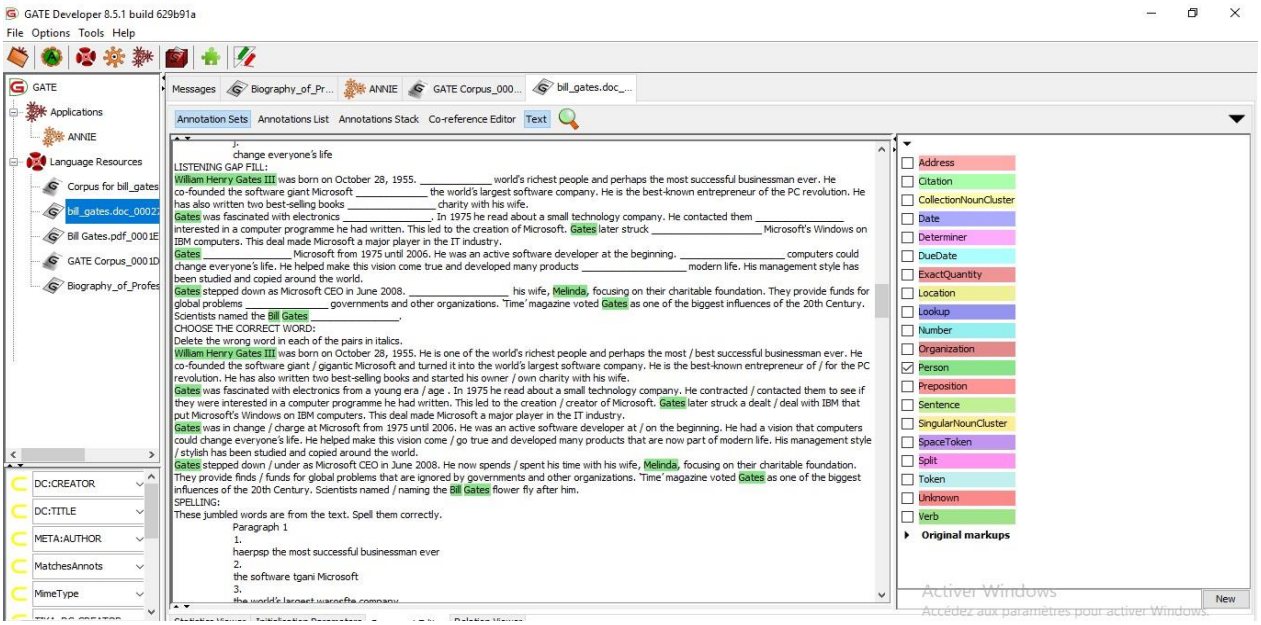


Figure 4.2 : Entités d'un seul type Personne sur un document de format DOC.

- Gate permet de reconnaitre dans son annotation les entités de type Organization :

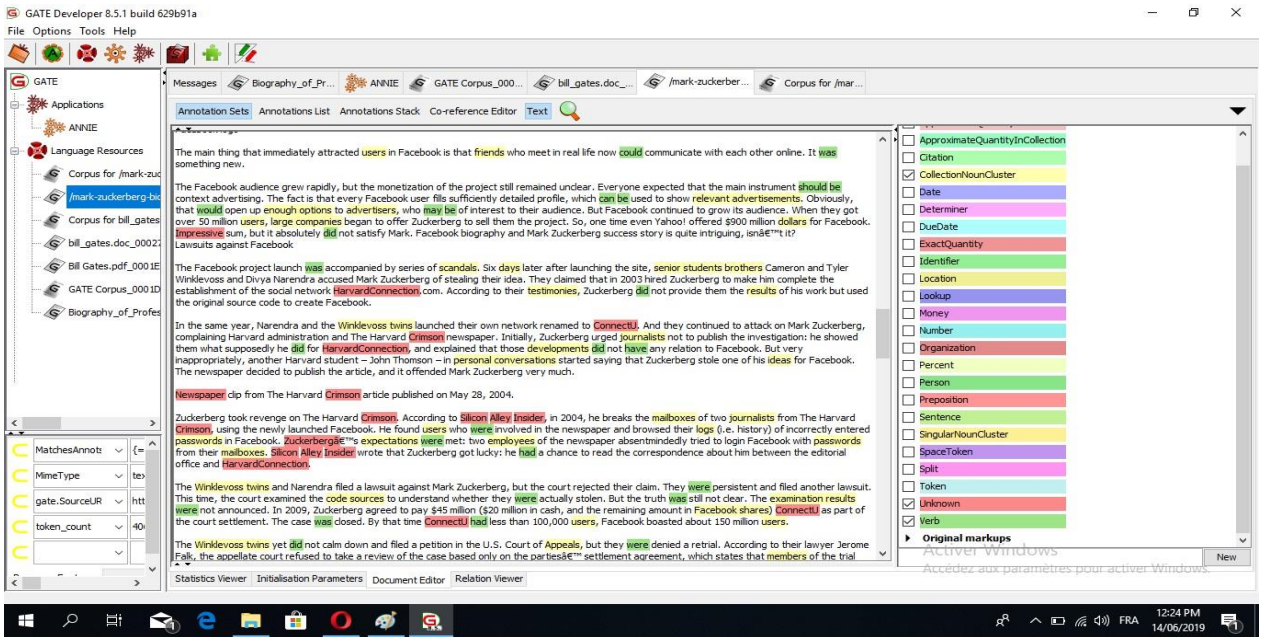
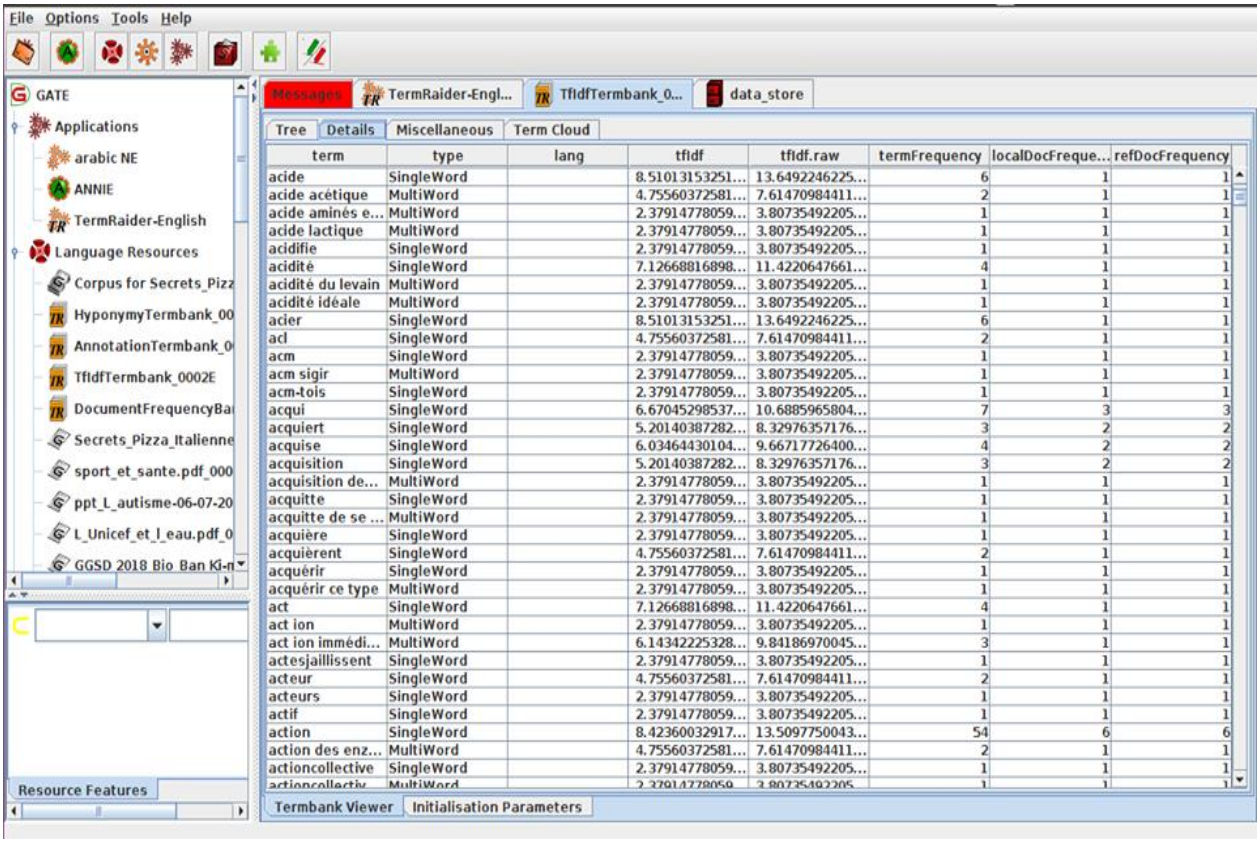


Figure 4.3 : Entités de différents types (CollectionNounCluster, Unkown, Verb) sur un document de format HTML.

CHAPITRE 4 : IMPLEMENTATION

Après l'annotation, nous avons sauvegardé les documents annotés sur le fichier gate.xml et sur les datastores spécifiques à GATE : LuceneDataStores qui font l'indexation des documents automatiquement.

Ensuite, nous avons utilisé l'application TermRaider pour calculer les tf/idf, son résultat peut être sauvegardé pour être utilisé pour la classification des documents par pertinence.



term	type	lang	tfIdf	tfIdf.raw	termFrequency	localDocFreque...	refDocFrequency
acide	SingleWord		8.51013153251...	13.6492246225...	6	1	1
acide acétique	MultiWord		4.75560372581...	7.61470984411...	2	1	1
acide aminés e...	MultiWord		2.37914778059...	3.80735492205...	1	1	1
acide lactique	MultiWord		2.37914778059...	3.80735492205...	1	1	1
acidifie	SingleWord		2.37914778059...	3.80735492205...	1	1	1
acidité	SingleWord		7.12668816898...	11.4220647661...	4	1	1
acidité du levain	MultiWord		2.37914778059...	3.80735492205...	1	1	1
acidité idéale	MultiWord		2.37914778059...	3.80735492205...	1	1	1
acier	SingleWord		8.51013153251...	13.6492246225...	6	1	1
acl	SingleWord		4.75560372581...	7.61470984411...	2	1	1
acm	SingleWord		2.37914778059...	3.80735492205...	1	1	1
acm sigir	MultiWord		2.37914778059...	3.80735492205...	1	1	1
acm-tois	SingleWord		2.37914778059...	3.80735492205...	1	1	1
acqui	SingleWord		6.67045298537...	10.6885965804...	7	3	3
acquiert	SingleWord		5.20140387282...	8.32976357176...	3	2	2
acquise	SingleWord		6.03464430104...	9.66717726400...	4	2	2
acquisition	SingleWord		5.20140387282...	8.32976357176...	3	2	2
acquisition de...	MultiWord		2.37914778059...	3.80735492205...	1	1	1
acquitte	SingleWord		2.37914778059...	3.80735492205...	1	1	1
acquitte de se ...	MultiWord		2.37914778059...	3.80735492205...	1	1	1
acquière	SingleWord		2.37914778059...	3.80735492205...	1	1	1
acquièrent	SingleWord		4.75560372581...	7.61470984411...	2	1	1
acquérir	SingleWord		2.37914778059...	3.80735492205...	1	1	1
acquérir ce type	MultiWord		2.37914778059...	3.80735492205...	1	1	1
act	SingleWord		7.12668816898...	11.4220647661...	4	1	1
act ion	MultiWord		2.37914778059...	3.80735492205...	1	1	1
act ion immédi...	MultiWord		6.14342225328...	9.84186970045...	3	1	1
actesjaillissent	SingleWord		2.37914778059...	3.80735492205...	1	1	1
acteur	SingleWord		4.75560372581...	7.61470984411...	2	1	1
acteurs	SingleWord		2.37914778059...	3.80735492205...	1	1	1
actif	SingleWord		2.37914778059...	3.80735492205...	1	1	1
action	SingleWord		8.42360032917...	13.5097750043...	54	6	6
action des enz...	MultiWord		4.75560372581...	7.61470984411...	2	1	1
actioncollective	SingleWord		2.37914778059...	3.80735492205...	1	1	1

Figure 4.4 : Calcul de Term Frequency dans un corpus.

TermRaider permet aussi de calculer le nombre de termes dans un document :

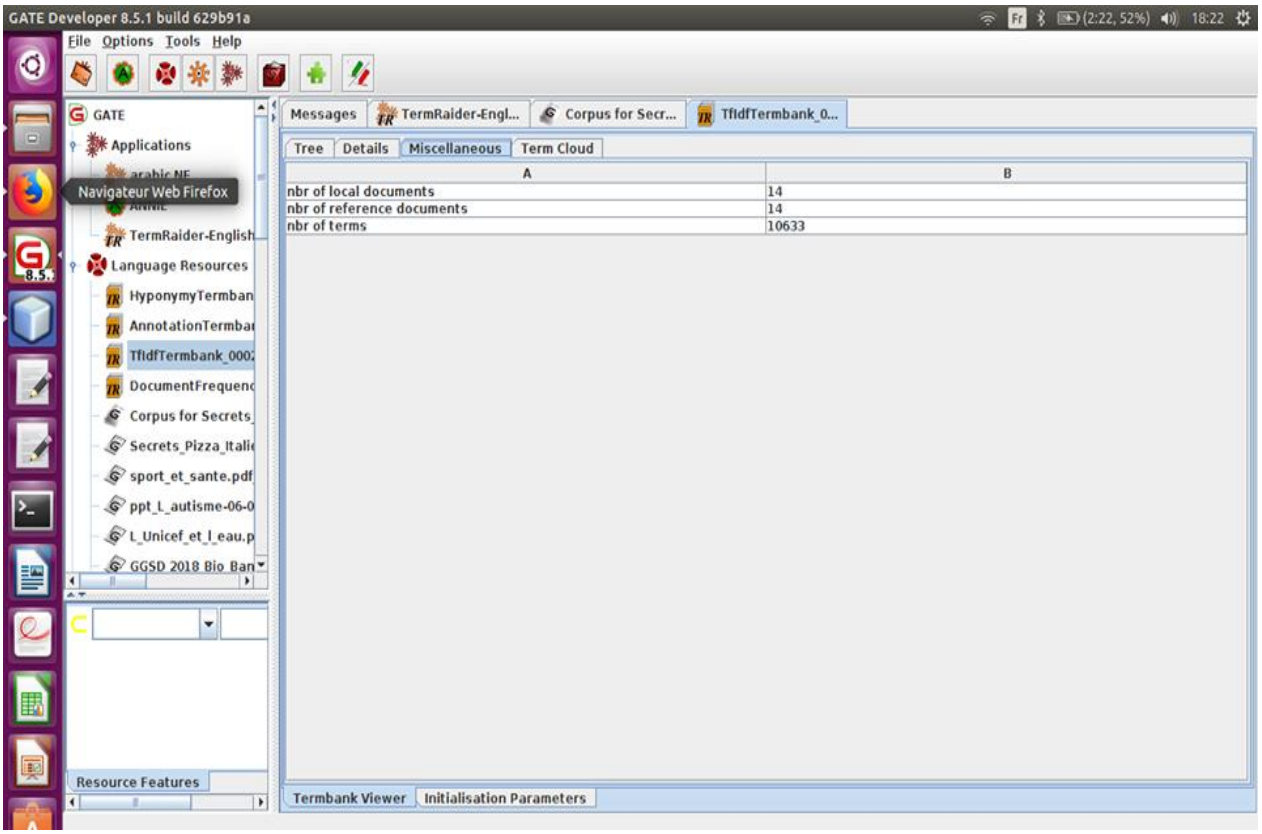


Figure 4.5 : Calcul de statistiques (nombre de termes) dans un corpus.

Nous avons implémenté le reste du système sur l'environnement NetBeans comme suit :

- Nous avons commencé par l'interface d'accueil défini dans la figure (figure 4.6) :
- Nous avons implémenté l'interface de recherche pour que l'utilisateur puisse taper sa requête :

CHAPITRE 4 : IMPLEMENTATION

Figure 4.6 : Interface d'Accueil de notre système

Notre système fournit à l'utilisateur une interface graphique qui lui permet d'interagir avec notre système et de faire une recherche et d'obtenir les documents qui correspondent à sa requête, parmi nos interfaces : l'interface Accueil défini dans la figure suivante :

CHAPITRE 4 : IMPLEMENTATION

Figure 4.7 : Interface de recherche de notre système

Finalement, en cliquant sur l'icône de recherche, l'utilisateur va avoir une autre interface où il va avoir comme résultat de sa recherche plusieurs documents classés par pertinence, ces derniers concernent sa requête (figure 4.8.)

CHAPITRE 4 : IMPLEMENTATION

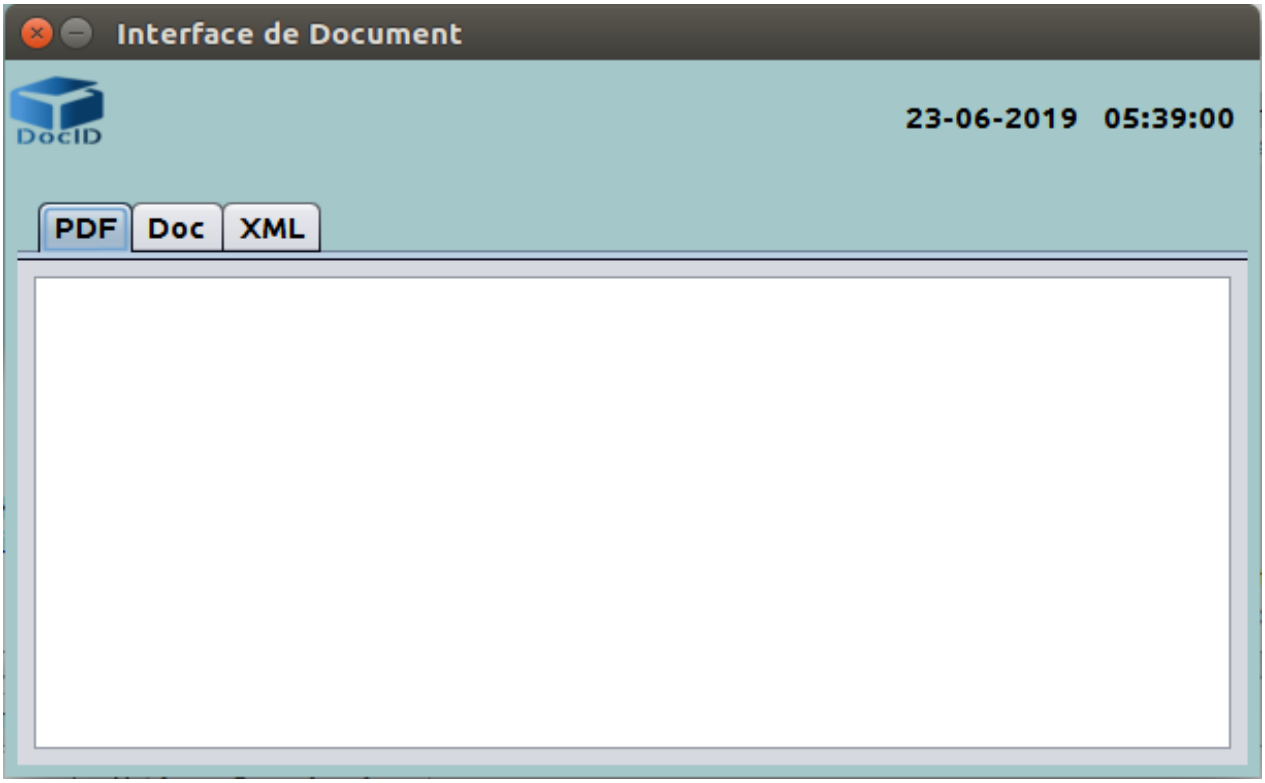


Figure 4.8 : Interface des documents résultants

IV.4 Conclusion :

Nous avons vu dans ce chapitre, les parties de développement de notre système de recherche d'information basé sur la détection des entités nommées avec une illustration de figures du logiciel GATE et du système lui-même.

CONCLUSION GENERALE

Ce travail de mémoire de fin d'étude a été consacré à la recherche d'information basé sur les entités nommées. Les utilisateurs cherchent toujours à avoir le bon résultat dès la première recherche, ils n'aiment pas reformuler plusieurs fois leurs requêtes ou avoir chaque mot de sa requête à part, donc l'utilisation de la recherche par mot clé est une bonne méthode qui facilite la recherche de l'utilisateur.

L'objectif de notre travail est d'implémenter une approche sémantique par mot clé capable de retourner des documents pertinents aux utilisateurs qui cherchent de l'information sur le web pour produire à la fin un bon résultat conforme par un système de recherche d'information.

Ce travail est présenté dans ce mémoire en quatre chapitres :

Le premier est une étude de l'état de l'art dont le sujet est la recherche d'information (RI).

Le deuxième, est l'entité nommée (EN) qui est une partie du domaine de la recherche d'information.

Le troisième présente la modélisation et la conception de notre travail.

Le quatrième est une illustration du processus de notre travail.

Ce travail permettrait de passer le cap de la reconnaissance des données de type structuré dans des fichiers plats et de petites bases de données et d'ouvrir la voie à l'analyse et le traitement de fichiers plus complexes comme des textes non structurés (en améliorant Gate) et des images en important une application (Weka) qui est basée sur le Machine Learning.

REFERENCE

Bibliographie :

[1] Mémoire de master : « Un modèle de reformulation des requêtes pour la recherche d'information sur le web » (chapitre1). Auteur : Abbassi et Meftah. Date : 2013. Consulté le 04 décembre 2018.

[2] Mémoire de master : « Traitement automatique des langues pour l'accès au contenu des documents » (Chapitre 4). Auteur : Christian Jacquemin et PierreZweigenbaum. Date : 2000. Consulté le 23 décembre 2018.

[3] Mémoire de master : « Modélisation du processus de recherche d'information par les éléphants d'Asie sociaux ». Auteur : Djamel Mostefai et OmarFekir Date : Juin 2018. Consulté le : 07 décembre 2018.

[5]Mémoire de master :« Présentation (chapitre2) :Représentation de l'information (indexation) ». Auteur : Mohand Boughanem. Date : 2014. Consulté le : 19 janvier 2019.

[6] Thèse de doctorat : « Système de recherche étendue basé sur une projection multi-espaces ». Auteur : Hannech Amel. Date : juillet 2018. Consulté le : 21 janvier 2019.

[7] Thèse de doctorat : « Dispositif de recherche et de traitement de l'information en vue d'une aide à la constitution de réseaux d'entreprise. ». Auteur : Kafil Hadjlaoui. Date : 2013. Consulté le :29 janvier 2019.

[8] Thèse de doctorat : « Modèle de recherche d'information basé sur les réseaux bayésiens et Les réseaux possibilistes. ». Auteur : M. Kamel Garrouch. Date : 16 février 2017. Consulté le : 03 février 2019.

[9]Thèse de doctorat : « Un modèle vectoriel étendu de recherche d'informations adapté aux images ». Auteur : Jean Martinet, Yves Chiaramella et Phillipe Mulhem. Date : 22 décembre 2004. Consulté le 1 février 2019.

[10] Cours de master : « Cours 10 (Master 2 LFA) de TAL : Traitement Automatique des Langues (Université de Paris-Sorbonne). ». Date : 2012. Consulté le : 28 décembre 2019.

[13]Article scientifique : « Panorama des différents moteurs de recherches » Date : Février 2009. Consulté le 30 décembre 2018.

[14] Thèse de doctorat : « Recherche d'entités nommées complexes sur le Web – propositions

REFERENCE

pour l'extraction et pour le calcul desimilarité. ». Auteur : Armel FOTSOH TAWOFAING. Date : 27 février 2018. Consulté le 07 décembre 2018.

[15] Thèse de doctorat :« Détection de mots clés et d'expressions régulières en vue de la reconnaissance d'entités nommées dans des documents manuscrits. ». Auteur : Gautier Bideault. Date : 7 Mar 2016. Consulté le : 18 décembre 2018.

[16] Thèse de doctorat : « Contributions aux techniques de recherche d'informations. ». Auteur :SAIDI Imène. Date : 2014-2015. Consulté le : 18 janvier 2019.

[17] Article scientifique : « Sur le statut référentiel des entités nommées». Auteur : Thierry Poibeau. Date : 3 Octobre 2005. Consulté le 30 décembre 2018

[18] Thèse de doctorat : « extraction des entités nommées par projection cross-linguistique et construction de lexiques bilingues d'entités nommées pour la traduction automatique statistique. ». Auteur : Fatima Deffaf. Date : Mars 2015. Consulté le : 21 janvier 2019.

[19] Rapport de stage : « Extraction d'Entités nommées par les Graphes d'Unitex » Auteur : Tolone Elsa. Date : 2006. Consulté le : 15 février 2019.

[20]Thèse de doctorat : « Modèles graphiques discriminants pour l'étiquetage de séquences : Application à la reconnaissance d'entités nommées radiophoniques. ». Auteur : AzeddineZidouni. Date : 2010. Consulté Le : 25 janvier 2019.

[21] Thèse de doctorat : « Reconnaissance des entités nommées par exploration de règles d'annotation - Interpréter les marqueurs d'annotation comme instructions de structuration locale. ». Auteur : Damien Nouvel. Date : 14 Février 2013. Consulté le : 02 février 2019.

[22] Thèse de doctorat : « Reconnaissance des entités nommées dans des documents multimodaux». Auteur: Mohamed Hatmi. Date: 24 Mai 2015. Consulté le : 31 janvier 2019.

[23] Thèse de doctorat : « Acquisition de relations entre entités nommées à partirdecorpus ». Auteur : Mani Ezzat. Date : 6 Mar 2015. Consulté le : 09 février 2019.

[24] Article scientifique : « Extraction automatique d'entités et de relations par ontologies et

REFERENCE

programmation logique inductive ». Auteur : Bernard Espinasse, Rinaldo Lima, Diana Magdy. Date : Octobre 2016. Consulté le : 18 février 2019.

[26] Articlescientifique : « Developing LanguageProcessingComponentswith GATE » Auteur : Hamish Cunningham,Diana Maynard, Valentin Tablan, Cristian Ursu4, KalinaBontcheva. Date : 2001, Consulté le : 21 février 2019.

[27] Article scientifique : « Comparaison d'outils d'informatique linguistique pourl'extraction d'information ». Auteur : Khanh-Lam. Date : 5 janvier 2012. Consulté le :21 février 2019.

[28] Article scientifique : « Adaptation d'un système de reconnaissance d'entités nommées pour le français à l'anglais à moindre coût. Auteur : Mohamed HatmiDate : juin 2012. Consulté le : 22 février 2019.

Webographie :

[4] URL :https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203511-index definition/.Consulté le : 15 décembre 2018

[11] DéfinitionSeo. URL : https://www.definitions-seo.com/definition-du-tfidf/. Consulté le : 31 décembre 2018.

[12] SupInfo (InternationalUnivesity).URL : https://www.supinfo.com/articles/single/4435-differents-moteurs-recherches. Consulté le : 02 janvier 2019

[25] https://www.techopedia.com/definition/30348/open-calais. Consulté le 28 janvier 2019.

[29]https://gate.ac.uk/ Consulté le 28 février 2019

[30]https://www.java.com/fr/about/ Consulté le 4 juin 2019

[31]https://netbeans.org/index_fr.html Consulté le 7 juin 2019