



RÉPUBLIQUE DÉMOCRATIQUE DE MADAGASCAR
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE
LA RECHERCHE SCIENTIFIQUE
UNIVERSITÉ DE TOAMASINA



FACULTÉ DES SCIENCES ET TECHNOLOGIE (FST)
DÉPARTEMENT D'INFORMATIQUE
THÈSE DE MASTER

Domaine : Mathématique, Informatique et Applications

Filière : Informatique

Option : Génie Informatique

Présenté par : BOUDI Gislain Carino Rodrigue

Thème

**Moteur de recherche basé sur le modèle vectoriel pour
améliorer la recherche des thèses malagasy**

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

| | | | | |
|-------|-------------|---------------|---------------------------|-----------|
| Mme. | Prénom NOM | MC(B) | (Université de Laghouat) | President |
| Mlle. | Prénom NOM | MC(A) | (Université de Laghouat) | Examineur |
| Mr. | Prénom NOM | MC(B) | (Université de Laghouat) | Examineur |
| Mr. | Jérôme VELO | Professeur(B) | (Université de Toamasina) | Encadreur |

Année Universitaire 2022/2023

Résumé

Resumé

Remerciements

Remerciements

Table des matières

| | |
|---|-----------|
| Résumé | i |
| Remerciements | ii |
| Table des matières | ii |
| Introduction générale | 1 |
| Vue d'ensemble | 1 |
| Contexte de la recherche de thèses malagasy | 1 |
| Problématique | 2 |
| Objectifs du mémoire | 2 |
| Organisation du devoir | 2 |
| 1 Recherche d'information | 4 |
| 1.1 Introduction | 4 |
| 1.2 Pourquoi rechercher une information ? | 5 |
| 1.3 C'est quoi une information ? | 5 |
| 1.4 C'est quoi un document ? | 7 |
| 1.5 Classification des documents | 8 |
| 1.5.1 Type de classification des documents | 8 |
| 1.5.2 Algorithmes de classification global | 8 |
| 1.6 Besoin d'information | 9 |
| 1.7 Tâche de recherche de l'utilisateur | 9 |
| 1.7.1 Tache factuelle | 10 |
| 1.7.2 Tache interprétative | 10 |
| 1.7.3 Tache d'exploitation | 10 |
| 1.8 Modèle de recherche de l'utilisateur | 11 |
| 1.8.1 Ad-hoc : approche classique | 11 |
| 1.8.2 Filtering : approche moderne | 11 |
| 1.9 Facteur d'influence de la recherche d'information | 11 |
| 1.10 Processus de recherche d'information | 12 |
| 1.10.1 Introduction (Processus en U) | 12 |

| | | |
|----------|--|-----------|
| 1.10.2 | Première étape : Besoin d'information | 12 |
| 1.10.3 | Indexation des documents | 14 |
| 1.10.3.1 | C'est quoi un index ? | 14 |
| 1.10.3.2 | Approche d'indexation | 14 |
| 1.10.4 | Analyse lexical (Segmentation) | 15 |
| 1.10.4.1 | Ponctuation | 15 |
| 1.10.4.2 | Casse des lettres | 15 |
| 1.10.5 | Élimination des mots vides (Stop words) | 15 |
| 1.10.6 | Lemmatisation et Stemming | 16 |
| 1.10.7 | Normalisation | 16 |
| 1.10.8 | Sélection et pondération des termes | 17 |
| 1.10.9 | Étape d'appariement | 17 |
| 1.10.10 | Reformulation de la requête | 18 |
| 1.11 | Modèle de RI | 18 |
| 1.11.1 | Introduction | 18 |
| 1.11.2 | Caractéristique formel (DQFR) | 19 |
| 1.11.3 | Taxonomie de modèle | 19 |
| 1.11.4 | Catégorie de modèle | 19 |
| 1.11.5 | Modèle ensembliste | 20 |
| 1.11.5.1 | Modèle Booléen (Boolean Model) | 20 |
| 1.11.5.2 | Modèle Booléen Étendu (Extended Boolean Model) | 22 |
| 1.11.5.3 | D'autres modèles | 22 |
| 1.11.6 | Modèle algébrique | 22 |
| 1.11.6.1 | Modèle Vectoriel (Vector Space Model) | 22 |
| 1.11.6.2 | Autre modèle | 23 |
| 1.11.7 | Modèle probabiliste | 24 |
| 1.11.7.1 | Modèle probabiliste | 24 |
| 1.11.7.2 | Autres modèles | 25 |
| 1.12 | Mesure de similarité | 25 |
| 1.12.1 | Produit scalaire | 26 |
| 1.12.2 | Similarité cosinus | 27 |
| 1.12.3 | Similarité de Jaccard | 27 |
| 1.12.4 | Similarité de Dice | 27 |
| 1.13 | Conclusion | 27 |
| 2 | Système de recherche d'information | 28 |
| 2.1 | Introduction | 28 |
| 2.2 | Objectif et efficacité | 29 |
| 2.3 | Élément principaux | 29 |
| 2.3.1 | Base de données | 30 |
| 2.3.2 | Algorithme de classement | 30 |

| | | |
|----------|---|-----------|
| 2.4 | Catégorie de recherche | 30 |
| 2.5 | Type de moteur de recherche | 30 |
| 2.5.1 | Classique ou traditionnel | 31 |
| 2.5.2 | Sémantique ou moderne | 31 |
| 2.6 | Modèle adopté par un SRI | 31 |
| 2.7 | Évaluation d'un SRI | 31 |
| 2.7.1 | Évaluation fonctionnel | 32 |
| 2.7.2 | Évaluation de performance | 32 |
| 2.7.3 | Évaluation de performance de recherche | 32 |
| 2.7.3.1 | Précision | 33 |
| 2.7.3.2 | Rappel | 34 |
| 2.7.4 | Exemple de précision et rappel | 34 |
| 2.7.5 | Problème de précision et le rappel | 34 |
| 2.7.6 | Moyenne harmonique | 36 |
| 2.7.7 | Et d'autres mesures | 36 |
| 2.8 | Analyse du moteur de recherche existant | 36 |
| 2.8.1 | Thèses malgache en ligne | 36 |
| 2.8.2 | Bibliothèques universitaire en ligne | 37 |
| 2.8.3 | Google Scholar, theses.fr et d'autres | 39 |
| 2.9 | Conclusion | 39 |
| 3 | Traitement de Langage Naturel (TLN) | 40 |
| 3.1 | Introduction | 40 |
| 3.2 | Un peu d'histoire | 41 |
| 3.3 | Les niveaux de langage | 42 |
| 3.4 | Phonologie | 42 |
| 3.5 | Morphologie | 42 |
| 3.5.1 | Triage de document | 44 |
| 3.5.2 | Segmentation de texte | 44 |
| 3.5.3 | Catégories grammaticales | 46 |
| 3.6 | Analyse lexicale | 46 |
| 3.7 | Syntaxe | 47 |
| 3.7.1 | Analyse superficielle | 47 |
| 3.7.2 | Analyse en dépendance | 48 |
| 3.8 | Sémantique | 48 |
| 3.9 | Pragmatique | 49 |
| 3.10 | Approche du TLN | 50 |
| 3.10.1 | Approche symbolique | 50 |
| 3.10.2 | Approche statistique | 50 |
| 3.10.3 | Approche connexionniste | 50 |
| 3.11 | Application de TLN | 51 |

| | |
|--|-----------|
| 3.12 Conclusion | 51 |
| 4 Modèle vectoriel | 53 |
| 4.1 Introduction | 53 |
| 4.2 Problème de classification | 54 |
| 4.3 Méthodes de pondération des termes | 54 |
| 4.4 Variant du TF-IDF | 55 |
| 4.5 Document inversé | 55 |
| 4.6 Avantages | 57 |
| 4.7 Inconvénients | 57 |
| 4.8 Conclusion | 57 |
| 5 Simulation et Réalisation | 58 |
| 5.1 Approche utilisée | 58 |
| 5.1.1 UML | 58 |
| 5.2 Technologies utilisé | 58 |
| 5.2.1 Python | 58 |
| 5.2.2 MySQL | 59 |
| 5.3 Outils utilisée | 59 |
| 5.3.1 Visual Studio Code | 60 |
| Conclusion générale | 61 |
| Bibliographie | 63 |

Liste des figures

| | | |
|-----|---|----|
| 1.1 | Historique de la recherche d'information (DR. LOUNNAS BILAL, 2023) | 6 |
| 1.2 | Processus de base de la RI (MAZARI, 2022) | 7 |
| 1.3 | Processus en U de la RI (BELLAOUAR, DJOUDI et ZIDAT, 2009) | 13 |
| 1.4 | Taxonomie de modèle de RI (SOULIER, 2014) | 20 |
| 1.5 | Exemple de diagramme de requête de modèle Booléen (HIEMSTRA, 1999) | 21 |
| 1.6 | Représentation du document et la requête dans le modèle vectoriel (HIEMSTRA, 1999) | 23 |
| 1.7 | Diagramme de collection pour le terme social dans le modèle probabiliste (HIEMSTRA, 1999) | 24 |
| 2.1 | Matrice de contingence (ABU-SALIH, 2018) | 33 |
| 2.2 | Ensemble des documents (BAEZA-YATES et RIBEIRO-NETO, 1999) | 34 |
| 2.3 | Exemple de calcul de précision et rappel (BOUGHANEM, s. d.) | 35 |
| 2.4 | Courbe précision-rappel (BOUGHANEM, s. d.) | 35 |
| 2.5 | Statistique des documents (UNIVERSITÉ D'ANTANANARIVO, 2016) | 37 |
| 2.6 | Résultat de recherche (UNIVERSITÉ D'ANTANANARIVO, 2016) | 38 |
| 2.7 | Pagination (UNIVERSITÉ D'ANTANANARIVO, 2016) | 38 |
| 3.1 | Position du TLN par rapport à l'IA (DEVOPEDIA, Year) | 41 |
| 3.2 | Étape d'analyse de texte dans NLP (DALE, ROBERT ET MOISL, HERMANN ET SOMERS, HAROLD, s. d.) | 43 |
| 3.3 | Catégorie grammaticale (DALE, ROBERT ET MOISL, HERMANN ET SOMERS, HAROLD, s. d.) | 46 |
| 3.4 | Description morphosyntaxique (DALE, ROBERT ET MOISL, HERMANN ET SOMERS, HAROLD, s. d.) | 47 |
| 3.5 | Exemple de définition de la lexique <i>Link Grammar</i> (TANNIER, 2006) | 48 |
| 3.6 | Exemple d'analyse syntaxique avec <i>Link Grammar</i> (TANNIER, 2006) | 48 |
| 4.1 | Variant du TF (SAADOUNE, 2018) | 55 |
| 4.2 | Variant de l'IDF (SAADOUNE, 2018) | 56 |
| 4.3 | Variant du TF-IDF (SAADOUNE, 2018) | 56 |

Liste des tableaux

Introduction générale

Vue d'ensemble

La recherche d'information (RI) c'est le fait de rechercher une information dans une base de documents (Corpus) à travers un système de recherche d'information ou moteur de recherche. C'est aussi trouver un document pertinent qui satisfait un besoin d'information de l'utilisateur.

Un Système de Recherche d'Information (SRI) ou moteur de recherche est un ensemble de programmes qui stock les documents et permet de donner des résultats suivant un degré de pertinence en fonction d'une requête de l'utilisateur.

Le modèle vectoriel est un modèle mathématique utilisé dans la recherche d'information pour définir le comportement de la Système de Recherche d'Information et définit l'algorithme de similarité entre la requête de l'utilisateur et les documents ainsi que le classement des documents.

Pour qu'un étudiant puissent rédiger et travailler sur son travail de recherche, il est important de trouver les documents concernant le thème en question (Articles, Thèses, Mémoires, etc.) pour rédiger l'état de l'art ou cadre théorique. Il est donc nécessaire de consulter des ressources sur internet par le biais des moteurs de recherches ou ressources physiques (Documents imprimés).

Contexte de la recherche de thèses malagasy

Pour rechercher les thèses malagasy, les moteurs de recherche académique, géant comme Google et Google scholar n'est pas suffisamment efficace dans le sens qu'il y a pas beaucoup des base de documents (Thèses, Articles) malagasy en ligne et que ces moteurs n'arrivent pas à indexer ces documents dans leur base. Il faudra passer donc par un SRI local pour héberger ces documents ainsi que de les rechercher.

Actuellement, le premier système en place c'est « Thèse malgache en ligne » qui représente actuellement (Stock) 31160 documents provenant des six universités publiques dont 973 pour l'Université de Toamasina, 27789 pour l'Université d'Antananarivo, 673 pour l'Université de Toliara, 1190 pour l'Université de Mahajanga, 338 pour l'Université de Fianarantsoa et 197

pour l'Université d'Antsiranana (UNIVERSITÉ D'ANTANANARIVO, 2016).

Ces documents sont pas encore suffisant, vu le nombre d'étudiants qui obtient leur diplôme de Master. Ainsi que l'université d'Antananarivo occupe environ de 89% du nombre des documents totale et les autres le 11% pourcent restant. Il y a aussi des bibliothèques en ligne des universités mais la plupart sans interface pour faire des recherches par mots clés. Ces lacunes sont la source de motivation de ce mémoire pour apporter une amélioration.

Problématique

Alors l'identification et l'exploitation des thèses et mémoires malagasy dans un domaine spécifique d'un sujet de recherche sont encore un défi en raison de limitations des ces systèmes de recherche existants. Certains de ces systèmes sont souvent peu performants, difficiles à utiliser en raison d'une interface utilisateur peu conviviale, et limités en terme de ressources et limités aux ressources des six universités publiques malgaches (Si nécessaire). Par conséquent, il est difficile de mener une recherche approfondie et d'exploiter pleinement les résultats disponibles. De plus, ces travaux de recherche sont souvent peu visibles sur les systèmes de recherche d'information populaires tels que Google Scholar ou Mémoire Online. Cette situation limite l'exploitation et la valorisation des travaux pertinents réalisés par les étudiants et chercheurs malagasy. Et puis la sécurité de ces fruits de recherche est un véritable défi pour les SRI (Moteur de recherche) comme protection de droit d'auteurs, vols de compétence, quel serveur doit héberger ces documents pour éviter les vols de ces fruits de recherche, ...

Objectifs du mémoire

L'objectif de ce mémoire est de développer un système de recherche d'information, sous la forme d'un moteur de recherche, basé sur le modèle vectoriel, qui résout les problèmes rencontrés dans l'accès aux travaux de recherche malagasy tels que les mémoires et thèses. Ce système doit être convivial, performant et regrouper tous les documents provenant des établissements d'enseignement supérieur publics et privés. Ainsi ce système propose un accès plus sécurisé pour les documents pour éviter le vol et protéger le droits d'auteur par une mise en place de système d'authentification, et une système de paiement pour un document payant. Ce SRI doit être hébergé a Madagascar pour mieux protéger ces ressources. L'objectif principal est de permettre aux chercheurs malagasy d'accéder facilement aux travaux de recherche pertinents et de promouvoir la recherche locale avec une meilleur sécurité.

Organisation du devoir

Pour mieux élaborer ce thème, ce devoir va se dérouler en quatre partie. La première partie c'est l'état de l'art où on va analyser ou on en est actuellement sur la Recherche d'Information et la Système de Recherche d'Information ainsi que ses modèles. On va aussi voir la Traitement

de Langage Naturel (Natural Language Processing ou NLP). La seconde partie concerne la méthodologie où on va analyser et détailler la méthode vectoriel (Modèle Vectoriel ou Vector Space Model), ainsi que l'approche sémantique. Dans la quatrième partie on va voir l'étape de la réalisation de ce Système de Recherche d'Information ainsi que ces fonctionnalités et détails. Et la quatrième et dernière partie la conclusion et perspective.

Chapitre 1

Recherche d'information

1.1 Introduction

La recherche d'information est un domaine étendu et ancien qui trouve ses racines au début des années 1950, peu après l'invention de l'ordinateur (ZIANI, 2022). Plus précisément, elle date des années 1940 dès la naissance de l'ordinateur. Dans les années 1960 et 1970, la RI commence les expérimentations plus larges alors qu'au départ elle s'est concentré sur des applications des bibliothèques. Et puis l'intégration de l'IA dans la RI a partir des années 1980 ainsi que la naissance d'internet en 1990 propulse la RI et met en scène beaucoup d'applications. (SALTON, 1989)

La RI suscite donc un grand intérêt parmi les chercheurs, devenant de plus en plus attrayante avec l'avènement de machines puissantes et performantes en matière de stockage, faciles à utiliser et fiables (BLAIR et MARON, 1985). La recherche d'information est désormais une tâche récurrente pour les utilisateurs, que ce soit sur smartphones, ordinateurs, etc., car elle répond à leurs besoins quotidiens (BELLAOUAR, DJOUDI et ZIDAT, 2009). Ce domaine traite des données non structurées, principalement des données textuelles comme le langage naturel (BAEZA-YATES et RIBEIRO-NETO, 1999). Citons quelques définitions de la Recherche d'Information :

Definition 1.1.1. Selon Gérard Salton, père de la Recherche d'Information, la RI consiste à structurer, analyser, organiser, stocker, et rechercher de l'information. (BELLAOUAR, DJOUDI et ZIDAT, 2009)

Definition 1.1.2. Une activité de délivrer un ensemble des documents à l'utilisateur en fonction de son besoin d'information (BELLAOUAR, DJOUDI et ZIDAT, 2009).

Definition 1.1.3. C'est la Stockage, Gestion, Traitement, récupération des informations, documents ou site web pour satisfaire un besoin informationnel de l'utilisateur (ABU-SALIH, 2018).

Definition 1.1.4. La RI est un ensemble d'une base de documents et une phase de processus de recherche avec des méthodes, techniques permettant a l'utilisateur de récupérer un ou plusieurs documents (KIEN QUACHTAT, 2012).

Definition 1.1.5 (Système de Recherche d'Information). On appelle SRI ou moteur de recherche, un ensemble de programmes informatiques, qui met en œuvre des techniques et moyens pour trouver les documents pertinents afin de satisfaire le besoin d'information de l'utilisateur (ZIANI, 2022).

Voici un courte historique de la recherche d'information jusqu'au 1998 qui est l'arrivée de Google, illustré dans la Figure 1.1.

Le principe de base de la recherche d'information est illustré dans la Figure 1.2.

1.2 Pourquoi rechercher une information ?

Lorsqu'un utilisateur a un besoin informationnel (Besoin d'information) la RI est nécessaire pour satisfaire ce besoin. D'où l'initiation de sa Recherche d'information. C'est le besoin d'information de l'utilisateur qui déclenche alors le processus de RI qu'on va voir dans la section suivante.

Le défi de la recherche d'informations est de satisfaire le mieux la demande de l'utilisateur en retournant un ensemble des documents correspondant a ses demandes parmi un grand volume de documents (MAZARI, 2022).

1.3 C'est quoi une information ?

Une information est issu lorsqu'on donne un sens a un données. Un donnée est donc un unité élémentaire d'information, un octets constitué de bits c'est à dire des 0 et des 1. Un donnée est une représentation a laquelle une signification peut être rattaché, peut être quantitative ou qualitative et n'a pas de chance elle-même.

Une information est donc une collection des données pour donner une forme a une message que ce soit imagée, écrite ou orale pour réduire l'incertitude et transmettre quelque chose qui déclenche une action (BELLAOUAR, DJOUDI et ZIDAT, 2009), (BENAYACHE, 2005).

Selon JÉRÔME VELO (JÉRÔME VELO, 2009) une information est définie comme un fait, une réalité, qui augmente la connaissance des individus. Elle est constitué de *données* et de *sens* qui est attribué par un individu. Le sens dépend de l'individu qui reçoit la données.

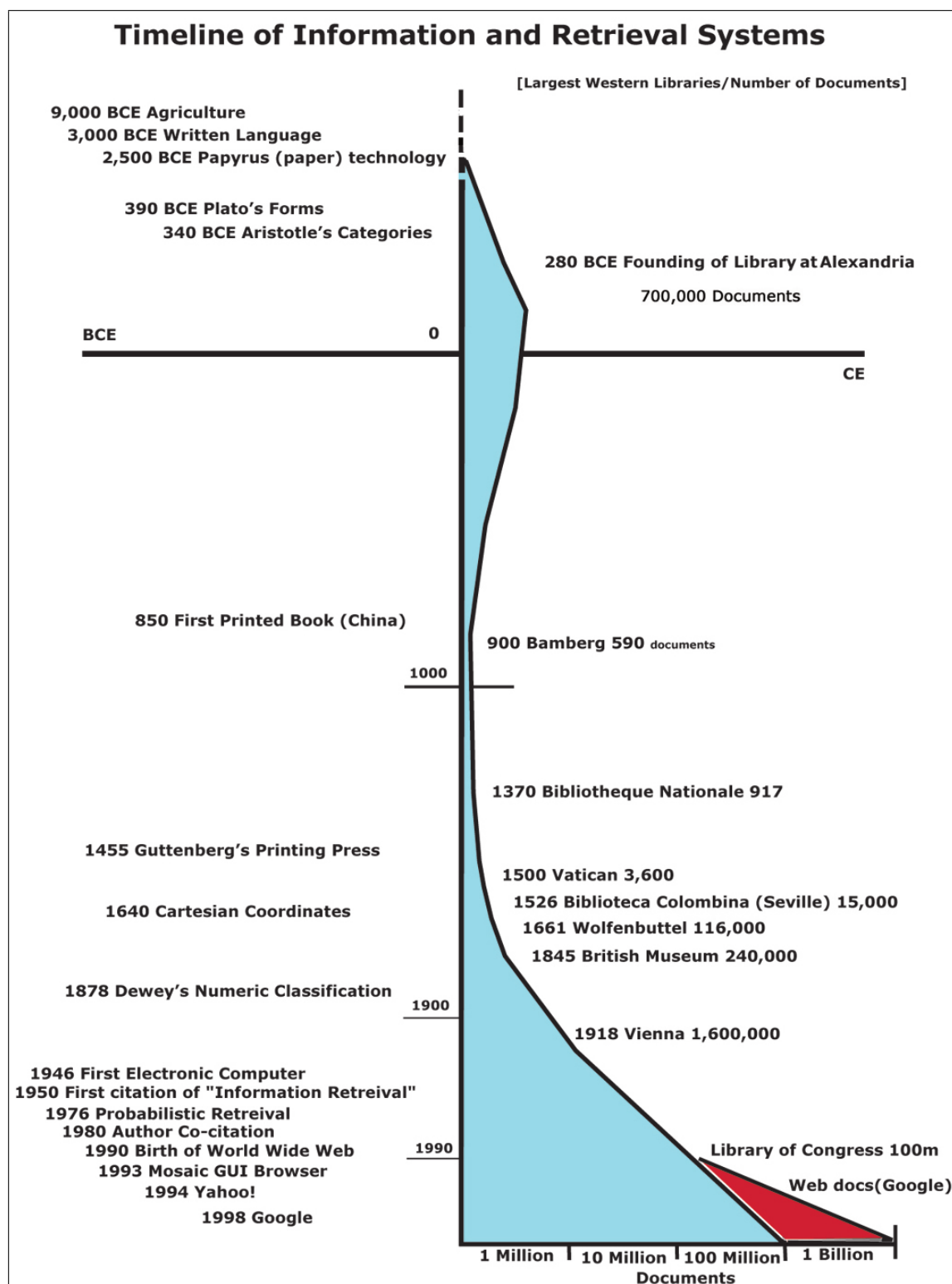


FIGURE 1.1 – Historique de la recherche d'information (DR. LOUNNAS BILAL, 2023)

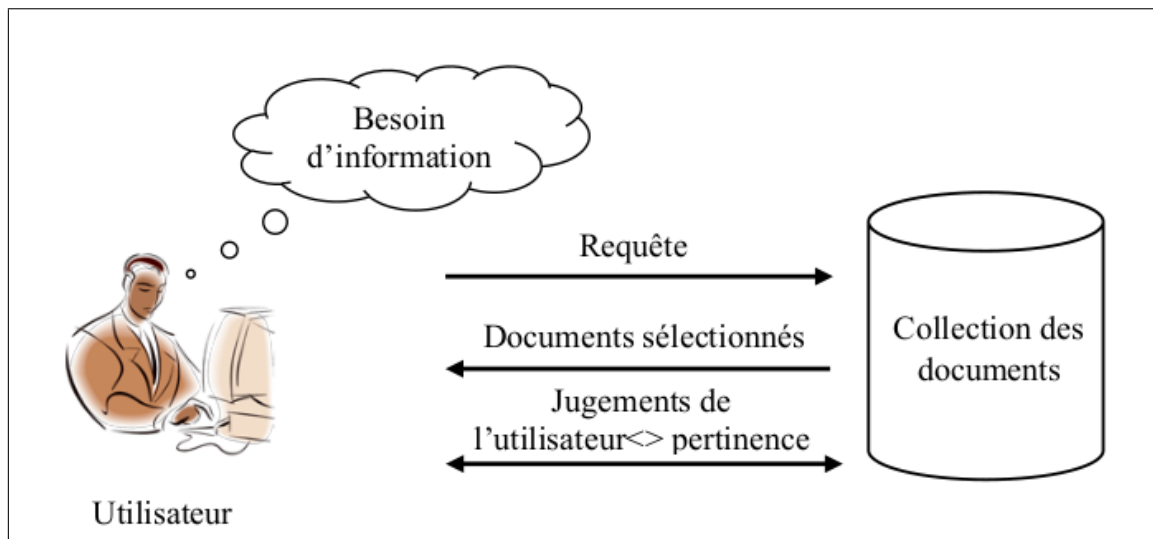


FIGURE 1.2 – Processus de base de la RI (MAZARI, 2022)

1.4 C'est quoi un document ?

BELLAOUAR, DJOUDI et ZIDAT propose quelques définitions d'un document, on va citer deux parmi ceux qui sont cité (BELLAOUAR, DJOUDI et ZIDAT, 2009).

Definition 1.4.1. Un document est généralement l'expression d'une pensée humaine.

Definition 1.4.2. Un document représente toute base de connaissance fixé matériellement, susceptible d'être utilisé pour la consultation, l'étude ou la preuve (imprimé, manuscrit, représentation graphique, ...).

Definition 1.4.3. On appelle document toute unité qui peut constituer une réponse à une requête d'utilisateur. Un document peut être un texte, un morceau de texte, une page Web (HTML), une image, une bande vidéo, sons, etc (SALTON, 1989).

Un document composé de cinq éléments (BELLAOUAR, DJOUDI et ZIDAT, 2009) :

- **Support d'enregistrement** : papier, disque magnétique, CD/DVD-ROM, etc. C'est un support où le document est stocké.
- **Forme d'enregistrement** : papier, codage ascii, codage vidéos, etc. C'est la forme de document dans le support.
- **Support de restitution** : papier, écran d'ordinateur, haut-parleurs, etc.
- **Forme physique de restitution** : écran sur papier, signal audio ou vidéos, etc.
- **Forme sémantique de restitution** : représentation qui respecte une certaine structure ou forme selon laquelle elle soit intelligible par le l'utilisateur. Écriture, sons, images, animés, etc.

Par la suite de ce recherche, nous traitons des documents textuelles, non structurés.

1.5 Classification des documents

1.5.1 Type de classification des documents

Pour pouvoir récupérer les documents qui sont similaires, ainsi que pour faciliter le filtrage des résultats par l'utilisateur, il est important de classer les documents.

La classification des documents est donc de grouper les documents similaires ensemble dans une classe. Cette opération se fait sur la collection des documents ou corpus. Il y a deux types de classification (BAEZA-YATES et RIBEIRO-NETO, 1999) tel que la classification *local* et la classification *global* :

- **Classification global** : les documents sont groupés selon leur occurrences dans la collection.
- **Classification local** : le groupement des documents est affecté par le contexte défini par la requête et les documents locaux sélectionnés. La classification varie selon la requête de l'utilisateur.

1.5.2 Algorithmes de classification global

Il y a beaucoup d'algorithmes qui permettent de classer les documents textuels, dont certains se basent sur l'algorithme du Machine Learning (ML) et de l'Intelligence Artificielle (AI). Voyons quelques algorithmes cités dans quelques revues scientifiques des classifications des documents textuels (B S HARISH, 2010 ; R MANIKANDAN, 2018).

Pour l'approche du machine learning, il y a des algorithmes qui se basent sur l'apprentissage supervisé, l'apprentissage non supervisé et l'apprentissage par renforcement.

- **K-Plus Proche Voisin ou K-Nearest Neighbor (KPP ou KNN)** : Est utilisé pour classer des textes. L'approche définit K classes pour classer. C'est une approche non paramétrique. Pour décider si d_i appartient à la classe C_k , la similarité ou la dissimilarité pour tous les documents d_j dans les données d'entraînement est déterminée.
- **Arbre de décision** : La décision est basée sur certaines conditions, et utilise un arbre. Les règles de classification sont représentées à travers le chemin du racine à la feuille. L'algorithme d'arbre de décision le plus connu est *ID3* et ses successeurs tels que *C4.5* et *C5*. Cette approche a des avantages comme, simple à expliquer et à comprendre par des personnes non connaisseuses du domaine, simple à mettre en place ainsi utile dans les analyses prédictives.
- **Naive Bayes Classifier** : Est un algorithme de machine learning, et généralement utilisé pour classer les documents WEB, qui est une approche probabiliste. Permet de catégoriser des documents, des news, etc. Cette algorithm est efficace, et qui nécessite moins de données d'entraînement, ainsi le résultat est efficace.
- **Support Vector Machine** : Est une approche supervisée du ML. Le SVM Classe les

données dans différentes classe par la recherche du ligne qui sépare les données d'entraînement en classes, on l'appelle ligne d'hyperplan. Il y a deux catégories tel que le *SVM* linéaire et le *SVM* non linéaire. Cette approche est performant pour classifier les données d'entraînement ainsi corrige bien la classification pour les futures données.

- **D'autres algorithmes** : Il y a encore d'autres algorithmes pour classifier les documents mais on va pas rentrer dans les détails dans le cadre de ce mémoire.

1.6 Besoin d'information

Definition 1.6.1. C'est la besoin d'information qui déclenche la recherche d'information de l'utilisateur comme cité ci-dessus. C'est une sensation qui porterait un individu a s'engager dans une activité de recherche d'information (BELLAOUAR, DJOUDI et ZIDAT, 2009).

Definition 1.6.2. C'est une abstraction mentale dont l'utilisateur a besoin pour répondre a une question ou demande particulière et qui est exprimé en langage naturel (MAZARI, 2022) (Reformulation nécessaire).

C'est cette besoin d'information que l'utilisateur doit traduire pour obtenir une requête afin qu'un SRI puisse satisfaire ce besoin, on dit souvent des mots clés de recherche (BAEZA-YATES et RIBEIRO-NETO, 1999).

La besoin d'information se catégorisent en trois types selon PARADIS (PARADIS, 1996) tel que :

- **Besoin vérificatif** : l'utilisateur recherche une donnée particulière, et savoir comment y accéder pour vérifier le texte avec des données qu'il possède déjà. En d'autre terme, l'utilisateur possède déjà les données ou la partie mais il a besoin de vérification. Le besoin ne change pas au cours de sa recherche, on dit que c'est *stable*.
- **Besoin thématique connu** : l'utilisateur cherche a éclaircir, revoir ou trouver des nouvelles informations concernant un sujet ou domaine connu. La besoin peut changer (*stable*) ou non (*variable*) au cours de sa recherche ainsi qu'il peut se raffiner. Le besoin peut s'exprimer de façon incomplète.
- **Besoin thématique inconnu** : l'utilisateur cherche des nouveaux concepts ou relations hors des domaines ou sujet connus. La besoin est variable et toujours exprimé de façon incomplète.

Pour mieux élaborer ce processus, on va analyser dans la section suivante la processus général de la recherche d'information, la processus en U.

1.7 Tâche de recherche de l'utilisateur

Une tache de recherche c'est qu'un utilisateur doit faire pour satisfaire un besoin d'information (BOUBÉE et TRICOT, 2010; KIEN QUACHTAT, 2012). Les taches se catégorisent

suivant deux types, une tâche fermée et une tâche ouverte. Une tâche fermée est une tâche dont l'utilisateur cherche une réponse exacte sur une requête donnée, tandis qu'une tâche ouverte est une tâche dont l'utilisateur cherche une réponse acceptable. Après certains états cités dans (KIEN QUACHTAT, 2012) et d'autres chercheurs, que la tâche ouverte utilise davantage la navigation, tandis que la tâche fermée privilégie les moteurs de recherche.

Il y a deux stratégies de recherche citées par YANG (YANG, 2000) en utilisant des requêtes :

- **Bottom-up ou mixtes** : la recherche commence par une sphère étroite (précise) puis l'élargit de plus en plus. L'utilisateur fait une recherche de plus précise à une recherche plus générale.
- **Top-down** : la recherche commence par une sphère large (générale) puis la rétrécit de plus en plus. L'utilisateur fait une recherche de plus générale à une recherche de plus en plus précise.

Selon KIEN QUACHTAT (KIEN QUACHTAT, 2012), il y a trois types de tâches :

- **Tâche factuelle** est une tâche généralement fermée
- **Tâche interprétative** est une tâche généralement ouverte
- **Tâche d'exploitation** est une tâche complètement ouverte

1.7.1 Tâche factuelle

Une tâche factuelle est catégorisée dans la tâche fermée, on peut le considérer comme une tâche fermée. Dans une tâche factuelle, les utilisateurs formulent des requêtes dans un SRI ou moteur de recherche, et lisent les résumés de chaque document dans la page de résultat. Ils veulent trouver une réponse sur la page de résultat et privilégier le résumé qui contient les mots clés, et en cliquant juste pour confirmation. Dans cette tâche les experts utilisent plutôt la stratégie bottom-up tandis que les novices utilisent la stratégie bottom-up (KIEN QUACHTAT, 2012).

1.7.2 Tâche interprétative

Une tâche factuelle est catégorisée dans la tâche ouverte, on peut le considérer comme une tâche ouverte. Les utilisateurs commencent par le parcours des informations générales, puis affiner leur recherche pour leurs besoins spécifiques. Ils privilégient la lecture de contenu des pages et la navigation, souvent une navigation interne à des pages web (via des liens internes) ou externes.

1.7.3 Tâche d'exploitation

Une tâche d'exploitation est une tâche complètement ouverte. Les utilisateurs se concentrent plus sur la page de résultat pour trouver un résultat qui semble correspondre au besoin, puis cliquent et naviguent dans la page pour explorer. Ils veulent trouver de l'information basée sur le contenu de la page. Dans cette tâche, les experts utilisent la stratégie top-down tandis que les novices utilisent la stratégie bottom-up (KIEN QUACHTAT, 2012).

1.8 Modèle de recherche de l'utilisateur

En général il y a deux approche de recherche d'information, dont l'approche classique qui est le Ad-hoc et et l'approche moderne qui est le Filtering (Filtre en français) (BAEZA-YATES et RIBEIRO-NETO, 1999).

1.8.1 Ad-hoc : approche classique

L'approche classique est l'approche la plus utilisée, d'où c'est une approche conventionnel de la Recherche de l'Information. Dans une approche classique, les documents sont resté relativement statique (Les documents sont préparés a l'avance). Puis arrivé ensuite la requête des utilisateurs pour trouver une réponses dans ces documents. En d'autre terme, les documents restent statique jusqu'à ce que la requête de l'utilisateur arrive.

1.8.2 Filtering : approche moderne

L'approche moderne consiste a mettre en place un profile complexe de l'utilisateur pour savoir quel documents ou informations pourra lui interrelié. Dans cette approche, les requêtes (préférences, type d'informations qui lui intéressent, ...) de l'utilisateur restent statique jusqu'à ce qu'un nouveau document arrive dans le système, et que ce système décide si c'est pertinent par rapport au profil de l'utilisateur. Cette approche est souvent utilisé pour les news, marketing, ...

1.9 Facteur d'influence de la recherche d'information

La recherche d'information de l'utilisateur est influencé par plusieurs facteurs, tel que l'expérience, facteur socio-culturel, et d'autres facteurs. On peut citer d'après l'analyse de ces facteurs fait par KIEN QUACHTAT (KIEN QUACHTAT, 2012) les facteurs ci-dessous :

- **Expérience de l'utilisateur** : l'expertise du domaine implique la qualité de choix des mots clés a utiliser, la stratégie de recherche ; l'expertise du système implique la connaissance de fonctionnement d'un SRI et la stratégie de la RI ; expertise RI implique la capacité d'utiliser un SRI spécifique.
- **Tache de recherche** : influencé par le type de tache de l'utilisateur, qu'il recherche de réponse exacte (*tache fermé*) ou recherche de réponse acceptable (*tache ouverte*).
- **Facteur socio-culturel** : la contexte sociale et la culture est indissociable de la recherche d'information ; le genre influence aussi la recherche, la stratégie de recherche utilisé par les femmes est différente de celle de l'homme, et la chance de réussite est différente ; l'âge aussi a son influence, les jeunes et les personnes âgés ont différente façon de faire une recherche. Par exemple, les jeunes on tendance a consulter moins de pages et formuler des requête plus souvent que les âgées.

- **Connaissance de fonctionnement de SRI** : parfois les connaissances sont fausses par rapport aux capacités des moteurs de recherche. L'utilisateur estime qu'il sait utiliser un SRI alors qu'il n'exploite véritablement les capacités du SRI.
- **D'autres facteurs** : d'autres facteurs sont aussi détaillés dans la section 2.2.5, comme la limite de temps, caractéristiques individuelles. Un tableau de synthèse de ces facteurs est présenté dans la section 2.3.

Pour plus de détails sur ces facteurs d'influences, voir « Recherche d'information sur le WEB et moteurs de recherche: le cas des lycéens », section 2.2

1.10 Processus de recherche d'information

Le modèle de processus le plus utilisé est le modèle en U. C'est un modèle qui traite les documents et la requête séparément et faire une appariement après.

1.10.1 Introduction (Processus en U)

La processus en U se déroule en général quatre étapes (BAEZA-YATES et RIBEIRO-NETO, 1999) et qu'il y a d'un côté le traitement des documents et de l'autre côté le traitement de la requête. L'indexation des documents : créer la base documentaire dans la base de données ; traitement et opérations sur la requête : traiter la requête de l'utilisateur, retirer les mots vides ; l'appariement de la requête et les documents : pour savoir quels documents répondent mieux à la requête de l'utilisateur ; Et enfin la reformulation de la requête de l'utilisateur : pour améliorer les résultats obtenus. La Figure 1.3 présente le processus de la recherche d'information en U.

Pour la suite, on va essayer d'expliquer en détails ce que font chaque partie de ce processus pour mieux élaborer l'étape de la recherche d'information.

1.10.2 Première étape : Besoin d'information

Comme définit dans la section 1.6, pour qu'une recherche d'information ait lieu, il faudra un besoin d'information. À ce stade la base documentaire est déjà mise en place et est déjà prête pour l'utilisation. L'utilisateur doit ensuite transformer son besoin en requête et obtenir des mots clés. Le système applique un traitement de texte sur la requête pour obtenir une requête finale qui sera utilisée par l'appariement. La requête pourra contenir un ou plusieurs mots clés (BAEZA-YATES et RIBEIRO-NETO, 1999).

Définition 1.10.1 (Requête ou Query). La requête est une interface entre l'utilisateur et le programme, formulée sous forme des mots-clés par l'utilisateur décrits en langage naturel (MAZARI, 2022).

Définition 1.10.2 (Requête ou Query). On appelle requête, l'expression du besoin d'information d'un utilisateur. Elle est en général exprimée sous forme des mots clés (SALTON, 1989).

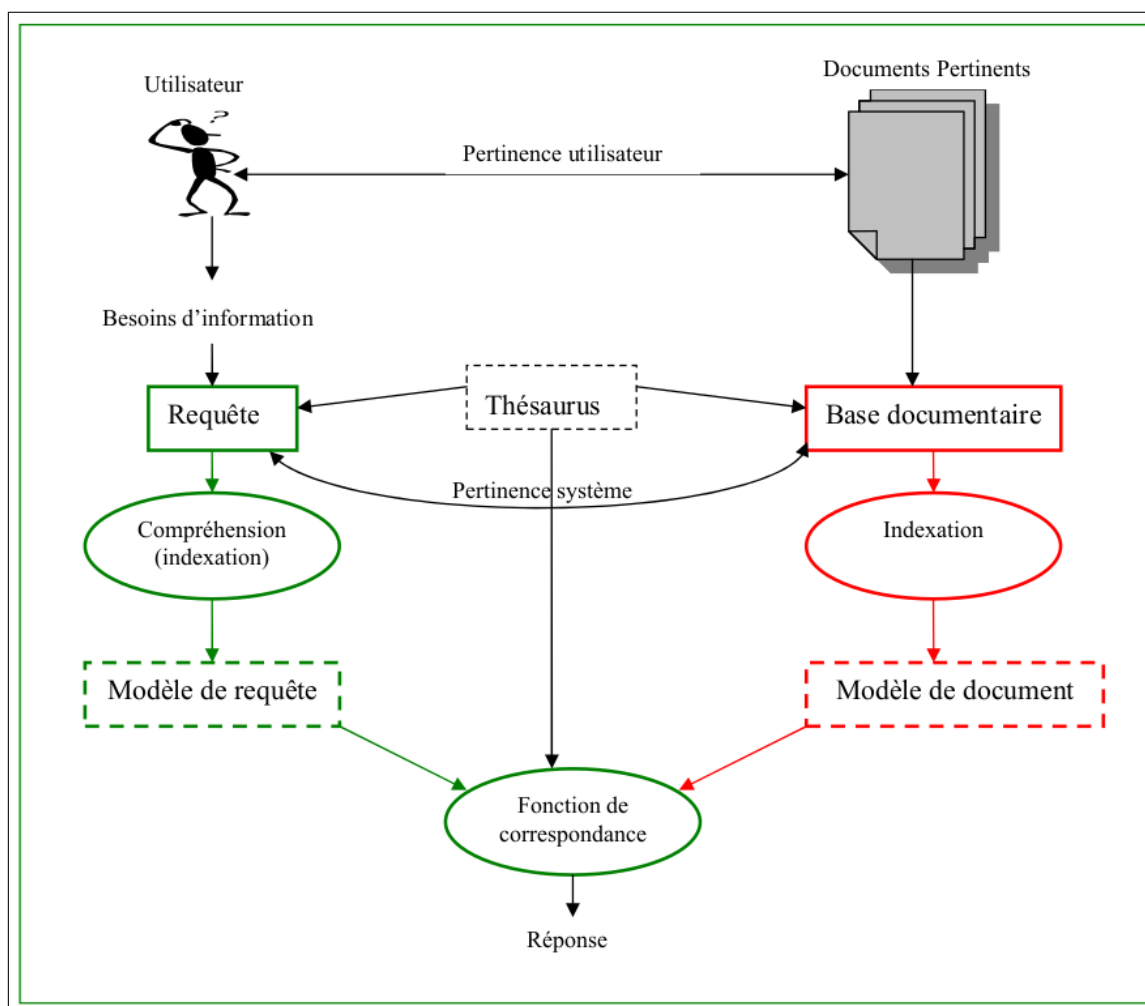


FIGURE 1.3 – Processus en U de la RI (BELLAOUAR, DJOUDI et ZIDAT, 2009)

1.10.3 Indexation des documents

Pour travailler efficacement avec les documents et pour pouvoir rechercher des informations dedans, il faut représenter les documents sous formes des mots clés que contient le document, sous l'approche d'indexation (YANG, 2000).

Definition 1.10.3. L'indexation ou tokenisation en anglais est l'action de représenter les contenu des documents sous forme d'index.

Definition 1.10.4. On appelle indexation ou processus d'indexation, le processus qui permet de construire les index a partir de l'analyse du documents tel que la vérification qu'un mot n'est pas dans un anti-dictionnaire (qui n'est pas un mot vide), et la lemmatisation ou racinisation pour trouver la racine des mots. Cette approche est formalisé en utilisant un langage d'indexation. (PARADIS, 1996).

1.10.3.1 C'est quoi un index ?

Un index est un mot ou groupe des mots sélectionné soit manuellement par un expert, soit automatiquement par un programme informatique (par un ordinateur) pour représenter un document. Les termes d'indexation (index terms) sont tous les index qui représentent un document. Un index est en général un nom ou groupe des noms.

Selon Salton, qui traite généralement les index comme représentant du document, Un index est une représentation synthétique de l'information relative a un document, qui met en évidence sa sémantique en vue d'une requête. Selon d'autres auteurs, se sont des mots clés (PARADIS, 1996).

1.10.3.2 Approche d'indexation

Comme on a évoqué précédemment et cité par dans (YANG, 2000) ainsi que dans (MAZARI, 2022), il y a en général quatre approche d'indexation tel que l'*indexation manuel*, l'*indexation intelligente*, l'*indexation automatique*, et l'*indexation basé sur les métadonnées*. Il y a une approche *semi-automatique* qui utilise a la fois l'indexation manuelle et automatique (une hybridation), cette approche consiste a créer les index automatiquement puis ces index sont validés par un expert (PARADIS, 1996).

- **Indexation manuelle (Human Indexing)** : indexation par un humain expert. Cette approche a une avantage d'être plus précise en sélection de terme d'indexation. Tandis qu'avec la croissance de volume d'informations, cette approche est de plus en plus obsolète, ainsi il se peut qu'il y a une incohérence entre les experts en sélectionnant les termes d'indexation. Cette approche est coûteux (BLAIR et MARON, 1985).
- **Indexation automatique (Automatic Indexing)** : c'est un programme informatique qui fait l'indexation. Cette approche a une avantage d'être moins coûteux qu'un indexation manuel, résout l'incohérence entre les termes d'indexation sélectionné. Mais peut être

moins efficace en sélectionnant les index, les index peuvent être moins précise. Cette approche utilise généralement la Traitement de Langage Naturel (NLP) et de la Statistique (Segmentation, Stop words removal).

- **Indexation intelligent (Intelligent Indexing / Agent-Based Indexing)** : Utilisation des robots (Spiders / Crawlers) pour l'indexation du WEB. Un robot ou spiders est un programme informatique qui parcourt les liens sur le web et index les documents trouvés. Cette approche a trois problèmes en général (Robot Standard Exclusion), les robots peuvent causer des surcharge au niveau des serveurs, invasives, et la mise à jour fréquent des sites web.
- **Indexation basé sur les métadonnées** : utilisation et interprétation des métadonnées, souvent utilisé dans la recherche d'information sur le web (HTML).

Dans ce document, on focalisera sur l'indexation automatique des documents généralement textuelle (PDF, Document WORD) par un programme informatique.

1.10.4 Analyse lexical (Segmentation)

L'analyse lexical des textes permet de faire une analyse de la lexique. Cette analyse peut varier suivant la langue utilisée. On analyse généralement les ponctuations, les espaces entre les mots et la casse des lettres. On considère que la séparation des mots est le caractère espace. L'analyse lexical est alors l'action de convertir les textes du document en groupe des mots (stream of words) qui peuvent être sélectionné comme termes d'indexation. Les nombres sont souvent une mauvaise index sauf s'il est mixé avec un mot. Les expressions régulières sont souvent utiles pour extraire des nombres comme le numéro de carte de crédit, ... (BAEZA-YATES et RIBEIRO-NETO, 1999 ; ZIANI, 2022).

1.10.4.1 Ponctuation

Les ponctuations sont normalement supprimés dans le cas d'analyse lexicales. Sauf dans le cas pour distinguer 'x.id' et 'xid.', la ponctuation ne doit pas être supprimé. (BAEZA-YATES et RIBEIRO-NETO, 1999)

1.10.4.2 Casse des lettres

Il faut convertir tous les mots en majuscules ou en minuscules. Mais en particulier, les langages de commande (ligne de commande) de Linux, car l'utilisateur ne veut pas convertir la casse des lettres (BAEZA-YATES et RIBEIRO-NETO, 1999). Ça peut perdre aussi l'aspect sémantique après la conversation.

1.10.5 Élimination des mots vides (Stop words)

On appelle un mot vide, un mot qui est plus souvent utilisé dans un document ou dans les termes de recherche d'un utilisateur comme les prépositions, les articles, les conjonctions.

Certains verbes, adverbes, et adjectifs pouvant être traité comme mots vides. Éliminer ces mots est important car un mot qui apparaît 80% dans le corpus n'est pas utile pour la recherche d'information. C'est l'action de supprimer les mots insignifiants (pronoms personnels, préposition, ...) (BAEZA-YATES et RIBEIRO-NETO, 1999 ; SAADOUNE, 2018 ; ZIANI, 2022).

Voici une liste non exhaustive des mots vides en français selon (RANKS NL, 2023) : le, de, du, mais, donc, car, ceci, cela, qui. A noter que pour un SRI Full-text qui prend tous les mots comme des termes d'indexations, il n'y a pas de suppression des mots vides (BAEZA-YATES et RIBEIRO-NETO, 1999).

1.10.6 Lemmatisation et Stemming

La lemmatisation ou racinisation de mot est une partie de le traitement de langage naturel (TLN) qui extrait la racine d'un mot pour regrouper les variants (biologiste, biologique -> biologie ; suis, est -> être). Cette approche réduit la taille des termes d'indexation (SAADOUNE, 2018 ; ZIANI, 2022).

Le Stemming regroupe aussi les variants d'un mot pour obtenir ce qu'on appelle un stem. Un stem est un portion d'un mot restant après la suppression de ses affixes (suffixes, préfixes), la suppression de pluriel, forme gérondif. Cette approche réduit aussi la taille des termes d'indexation. Le stemming ne permet pas de satisfaire la racinisation d'un mot (peut mal déterminer la racine), que certains moteurs de recherche n'adopte pas le stemming (BAEZA-YATES et RIBEIRO-NETO, 1999 ; Vaibhav SINGH et Vinay SINGH, 2022).

Le stemming possède plusieurs techniques, mais on va citer les techniques principaux qui sont :

- **Table lookup** : regarder le stem dans un table, mais qui est moins pratique en terme de stockage et accessibilité.
- **Porter algorithm** : est une technique simple, intuitive, peut être implémenté efficacement. Cette approche utilise la suppression des affixes.
- **Successeur** : est un approche plus complexe que la suppression des affixes. Se base utilisation des connaissances linguistiques structurés pour identifier les morphes.
- **N-Grams** : est une approche orienté plus vers la classification que stemming. Cette approche se base sur l'identification des diagrammes et trigrammes.

D'autres méthodes sont disponibles actuellement comme celle qui se base sur les expressions régulières, ...

1.10.7 Normalisation

La normalisation permet de classer les termes comme équivalent. Par exemple, le mot U.S.A doit être traité de la même façon que USA. Cette approche de normalisation peut amener a

des problèmes (SAADOUNE, 2018).

1.10.8 Sélection et pondération des termes

Comme on dit précédemment, pas tous les mots dans le documents sont significatifs. Sauf pour un moteur de recherche full-text qui index tous les termes. Et aussi pas tous les mots on une même degré ou poids d'importance dans un document. D'où la pondération des termes. Pour déterminer le poids d'un terme dans un collection des documents, on utilise en général deux approche : *Term Frequency (TF)* et *Inverse Document Frequency (IDF)*.

Le *TF (Term Frequency)* du terme t dans un document d est le nombre de fois où le terme t apparaît dans le document d . C'est la fréquence du ter t dans le document d . Plus ce fréquence est important, plus le terme est important pour représenter le document. Cette mesure sera normalisé afin d'éviter d'avoir un poids trop significatifs pour les documents avec beaucoup de contenu, on l'appelle *Normalized Term Frequency*.

L'*IDF (Inverse Document Frequency)* du terme t dans la collection des documents D est le nombre de document dans la collection contenant le terme t . C'est le nombre de document où le terme t apparaît dans la collection des documents D . Cette approche permet de déterminer l'unicité d'un terme au sein de la collection des documents. Un terme est important si moins de documents la contiennent ; un terme qui apparaît dans presque tous les documents n'est pas significatif, et peut être traité comme un mot vide.

La troisième approche qui est la combinaison des deux est la *TF-IDF*. La *TF-IDF* est la multiplication des deux valeurs obtenues précédemment ($TF * IDF$) pour obtenir le poids final du terme. Le terme est important s'il apparaît dans peu de documents (*IDF*) mais qu'il apparaît plusieurs fois dans le document qui le contient (*TF*).

Une fois ces termes sélectionné avec les poids correspondant, on crée un index inversé ou fichier inversé pour stocker ces termes. Cet index inversé va faire correspondre le document identifié par un ID avec les termes qui le compose avec le poids correspondant (ABU-SALIH, 2018 ; BAEZA-YATES et RIBEIRO-NETO, 1999 ; SAADOUNE, 2018 ; Vaibhav SINGH et Vinay SINGH, 2022). Tous ces notions va être détaillé un peu plus bas dans le document lorsqu'on abordera le modèle utilisé dans le cadre de ce mémoire.

1.10.9 Étape d'appariement

Pour pouvoir satisfaire le besoin de l'utilisateur, il faut apparier la requête de l'utilisateur et les documents (fichier inversé) en utilisant un algorithme d'appariement. Ces algorithmes permet de calculer la similarité entre un document et une requête et savoir quels sont les documents qui correspondent le mieux avec la requête avec un degré de similarité. Mathématiquement, c'est calculer la distance euclidien entre la requête et le document.

On calcule ces mesures pour déterminer la pertinence entre la requête de l'utilisateur et

les documents dans le corpus, on l'appelle *pertinence système*. Il y a aussi le jugement qui provient de l'utilisateur en vue des résultats qu'il obtient par rapport à sa demande, on l'appelle *pertinence utilisateur* (MARTINET, 2004).

La notion de pertinence est très complexe. En général un document est pertinent si l'utilisateur trouve les informations qui satisfont ses besoins dans le document. C'est sur ce notion de pertinence que le SRI doit juger la pertinence. (SALTON, 1989)

Les documents trouvés sont retournés à l'utilisateur généralement suivant un degré de pertinence décroissant par rapport à la requête. Certains modèles ne possèdent pas ce classement des résultats comme le modèle Booléen qu'on va voir dans la section précédente, ainsi que décortiquer ces mesures de similarité.

1.10.10 Reformulation de la requête

La requête de l'utilisateur est parfois courte ou incomplète, alors il faut reformuler la requête par l'ajoute, modification ou suppression des mots clés dans la requête. Il est aussi possible de modifier le poids des termes dans la requête. Parmi les différentes approches proposées, MAZARI (MAZARI, 2022) note trois approches utilisées dans la reformulation de la requête tel que :

- **Thésaurus** : cette approche utilise une base de connaissance linguistiques pour améliorer la requête de l'utilisateur.
- **Basé sur les co-occurrences** : utilisent le calcul basé sur des co-occurrences des termes.
- **Relevance feed-back** : cette approche se base sur la reformulation de la requête par l'utilisateur par rapport aux pertinences des documents qu'ils reçoivent. En rajoutant, modifiant ou enlevant des termes de recherche ou mots clés.

1.11 Modèle de RI

1.11.1 Introduction

Il y a deux raisons pour avoir un modèle dans la recherche d'information. La première, un modèle guide la recherche et donne sens à la discussion académique. La seconde, un modèle sert comme un plan (blueprint) modèle pour implémenter un système de recherche d'information (HIEMSTRA, 1999).

Un modèle sert à décrire le processus computationnel du RI (comment les documents sont stockés, comment sont stockés les index, etc.), décrire le processus humain (besoin d'information, interaction), ainsi la définition implicite ou explicite de la pertinence (A. ROZENKNOP, s. d.).

Pour bien formaliser la recherche d'information, il est nécessaire de définir des modèles à utiliser. L'étape d'appariement dans la section Étape d'appariement, nécessite un modèle.

1.11.2 Caractéristique formel (DQFR)

Definition 1.11.1. Un modèle de recherche d'information est quadruplet DQFR tel que (BAEZA-YATES et RIBEIRO-NETO, 1999 ; Vaibhav SINGH et Vinay SINGH, 2022) :

- **D** : ensemble des représentation (vue logiques) des documents dans la collection
- **Q** : ensemble des représentation (vue logiques) des besoins de l'utilisateur. On l'appelle requête ou query en anglais.
- **F** : framework pour modéliser les documents, la requête ainsi que leurs relations.
- **R** (q_i, d_j) : fonction de classement qui associe un nombre réel avec la requête q_i appartient a Q et le document d_j appartient a D . Cette classement permet d'ordonner les résultats selon leur pertinence.

1.11.3 Taxonomie de modèle

Selon BAEZA-YATES et RIBEIRO-NETO il y a quatre familles principales des modèles de RI qui est illustré dans la Figure 1.4 :

- **Les modèles basé sur les textes du document** : les modèles de RI classique (*Théorie des ensembles, Algébrique, Probabiliste*), et les modèles basé sur le texte semi-structuré (*Texte semi-structuré*)
- **Les modèles basé sur le liens entre documents** : les modèles orienté web (*PageRank, Hubs et autorités*)
- **Les modèles basé sur les documents multimédia** : les modèles de recherche d'images, vidéos, musiques, etc.

Dans le cadre de ce mémoire, on va analyser la famille des modèles classiques basé sur les textes du document, car ce qui concerne mieux notre étude.

1.11.4 Catégorie de modèle

Comme énoncé précédemment, on va s'intéresser dans la famille des modèles classiques basé sur les textes du document. Dans cette famille, on a (BAEZA-YATES et RIBEIRO-NETO, 1999 ; SOULIER, 2014 ; ZIANI, 2022), il y a trois grandes catégories des modèles : le modèle **théorique ou ensembliste** (theoretic), le modèle **algébrique** (algebraic) et le modèle **probabiliste** (probabilistic).

Dans le modèle ensembliste, on trouve principalement le modèle **Booléen**, le modèle **Booléen étendue**. Dans le modèle algébrique, on a principalement le modèle **vectoriel** (Vector Space Model), le modèle de **réseau de neurone** (Neural Network Models), et d'autres modèles. Dans le modèle probabiliste, on trouve principalement le modèle probabiliste (Probabilistic model), le **réseau d'inférence** (Inference Network) et d'autres modèles.

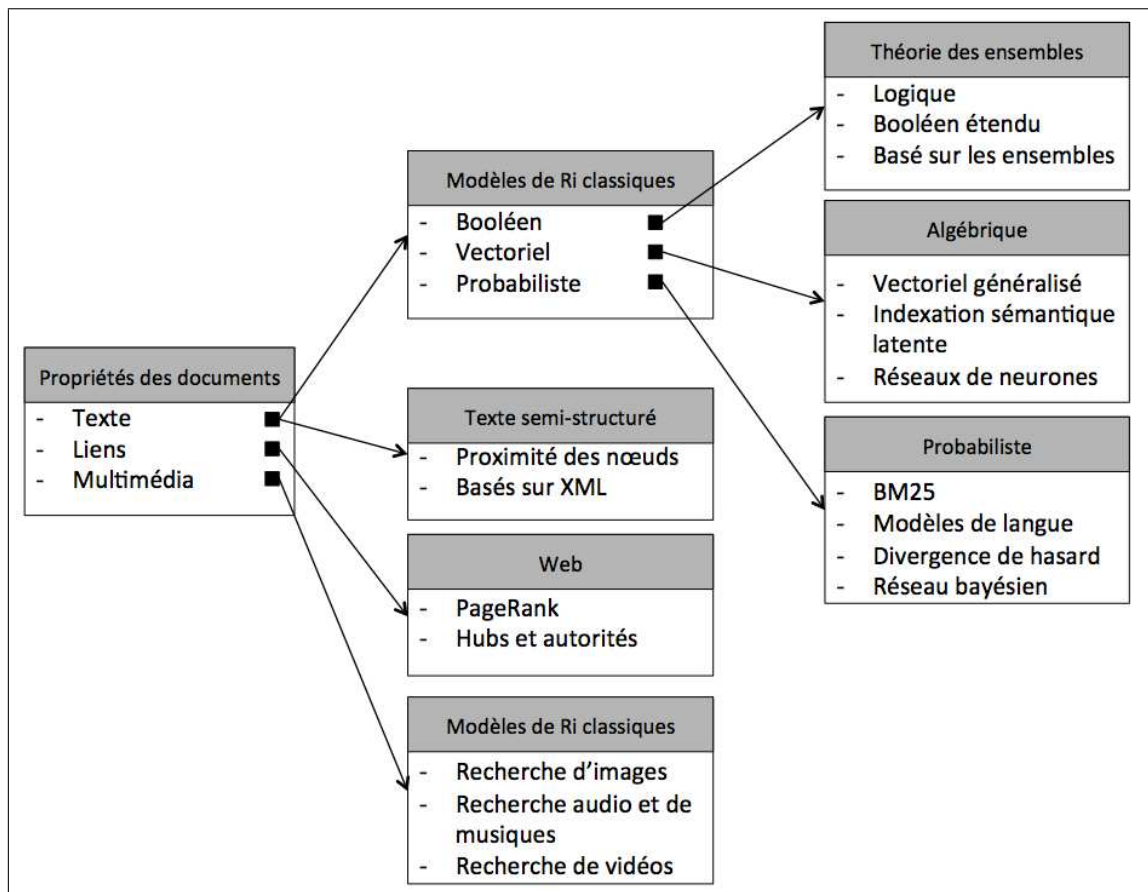


FIGURE 1.4 – Taxonomie de modèle de RI (SOULIER, 2014)

Ces modèles sont des modèles de recherche, mais il y a aussi des modèles pour la navigation (Browsing). Pour plus de détails (BAEZA-YATES et RIBEIRO-NETO, 1999).

1.11.5 Modèle ensembliste

Dans le modèle ensembliste, il y a quelques modèles, mais dans le cadre de ce mémoire, on va voir les deux principaux.

1.11.5.1 Modèle Booléen (Boolean Model)

Un exemple de requête dans le modèle Booléen est illustré sur la Figure 1.5.

Ce un modèle de base dans la recherche d'information. Ce modèle est le premier modèle utilisé dans la recherche d'information. Dans ce modèle, les documents et les requêtes sont représentés comme des mots clés (ensemble des mots). Ce modèle est simple, facile à mettre en place et utilise l'algèbre de Boole. La stratégie de recherche est basé sur la décision binaire. Ce modèle a possède un avantage d'être facile à mettre en place, simple et exacte au niveau des résultats. Ce pour cette exactitude que ce modèle est aussi un modèle exacte (Exact Match Model).

Les termes sont pondérés de façon binaire.

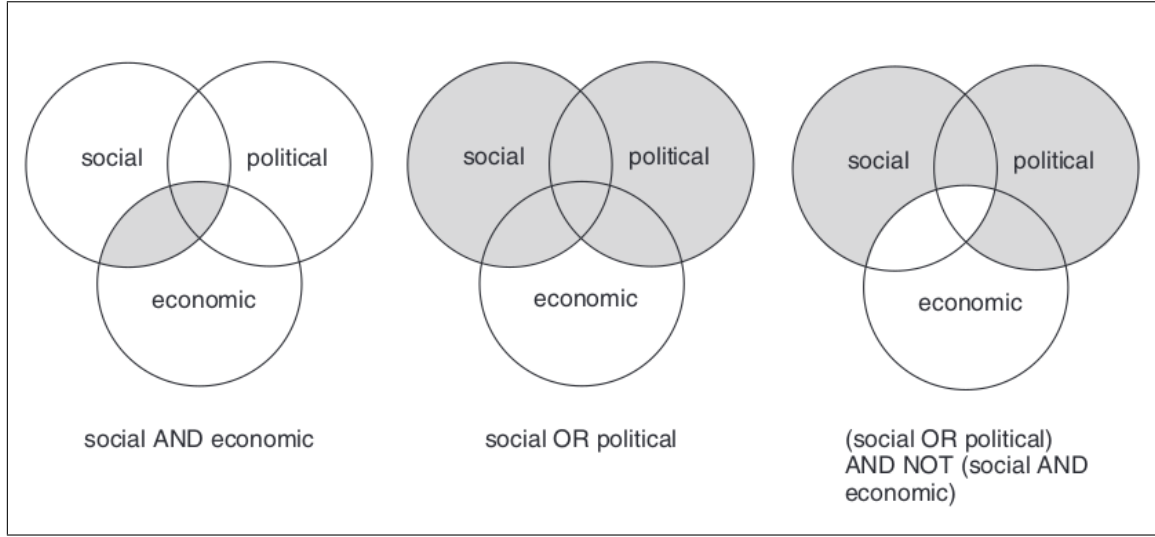


FIGURE 1.5 – Exemple de diagramme de requête de modèle Booléen (HIEMSTRA, 1999)

Tous les termes d'indexation ont la même poids, soit il existe dans le document soit non. S'il existe dans le document, son poids est 1, sinon 0. La similarité entre q et d_j est égal a 1 si le document terme d_j valide l'expression booléen q , sinon 0. Un document est donc jugé pertinent par le système s'il valide l'expression booléen de la requête.

Dans ce modèle, la requête est exprimé en utilisant les opérateurs booléens ET (AND), OU (OR) et NON (NOT).

Par contre, ce modèle a quelques inconvénients, il est difficile de transformer la besoin d'information en expressions booléen surtout pour un utilisateur lambda (un utilisateur qui ne sait absolument rien de l'expression booléen) donc la requête a tendance un peu simplifié ou même incomplète ce qui affecte la qualité des résultats. Et puisque la décision est binaire, et que le poids soit 1 soit 0, il n'y a pas de classement possible pour les documents retournés. L'utilisateur est alors obligé de parcourir les résultats pour juger la pertinence. Puisque le modèle est exacte, le modèle ne permet pas de sélectionner les documents qui peuvent intéressé l'utilisateur et qui peuvent être pertinent par rapport a ses requête, et qu'il est impossible de récupérer une réponses partielles qui valide certains expression de la requête. (BAEZA-YATES et RIBEIRO-NETO, 1999; SOULIER, 2014).

Definition 1.11.2 (Modèle booléen). On note $w_{ij} \in \{0, 1\}$ le poids du terme numéro i dans le document numéro j qui est binaire \vec{q}_{dnf} , forme normal de disjonction de la requête q . Considérons \vec{q}_{cc} composant conjonctive de \vec{q}_{dnf} . La similarité du document d_j par rapport a la requête q est définie par :

$$Sim(d_j, q) = \begin{cases} 0 & \text{Sinon} \\ 1 & \text{S'il } \exists \vec{q}_{cc} \mid (\vec{q}_{cc} \in \vec{q}_{dnf}) \wedge (\forall k_i, g(\vec{d}_j) = g_i(\vec{q}_{cc})) \end{cases}$$

Si $Sim(d_j, q) = 1$, le document d_j est pertinent a la requête q ; dans le cas contraire, le

document n'est pas pertinent par rapport a la requête.

1.11.5.2 Modèle Booléen Étendu (Extended Boolean Model)

En vue de ces inconvénients, le modèle booléen est amélioré afin de résoudre certains problèmes avec le modèle Booléen. Premièrement, le modèle Booléen étendue a mis en place la possibilité de faire une similarité partiel et la pondération des termes. Il est étendu en ajoutant la fonctionnalité du modèle vectoriel qui est la combinaison de formulation de requête avec les caractéristiques du modèle vectoriel. Ce modèle est introduit par Salton, Fox, et Wu en 1983. Mais ce modèle n'a pas été largement utilisé (BAEZA-YATES et RIBEIRO-NETO, 1999).

1.11.5.3 D'autres modèles

Il y a aussi d'autres modèles dans la catégorie ensembliste, tel que le modèle de Fuzzy (Fuzzy Set Model, Region Models), ..., mais qui ne sont pas détaillé dans le cadre de ce mémoire. Pour plus de détails, voir chapitre 2 (BAEZA-YATES et RIBEIRO-NETO, 1999) et chapitre 1 (HIEMSTRA, 1999).

1.11.6 Modèle algébrique

Dans cette catégorie, on va analyser le modèle vectoriel principal.

1.11.6.1 Modèle Vectoriel (Vector Space Model)

Ce modèle a un alternative qui est le Modèle Vectoriel Généralisé par Salton (SALTON, GERARD ET WONG, ANITA ET YANG, CHUNG-SHU, 1975), qui se base sur des vecteurs. Les documents et les requêtes sont représentés par un vecteur dans un t-espace vectoriel (représentation algébrique). Ce modèle accorde une appariement partielle, et les termes ne sont pas pondérés de manières binaires. Les pondérations des termes le plus souvent utilisé sont la TF, IDF, et la TF-IDF (SAADOUNE, 2018).

Un score de similarité est alors exprimé comme un mesure de proximité entre deux entités correspondant a l'angle qui sépare les deux vecteurs. On va analyser brièvement ces mesures dans la section 1.12.

Le modèle vectoriel a certains avantages : Facile a mettre en œuvre, possibilité d'avoir un appariement approximatif avec un degré de pertinence (L'utilisateur reçoit des documents qui pourrait lui intéressé), possibilité d'organiser les résultats suivant leur pertinence (L'utilisateur passe moins de temps a filtrer les résultats puisqu'ils sont déjà ordonné), ainsi ce modèle peut définir une limite pour la mesure de similarité et de n'afficher que les documents qui sont en dessus de cette limite pour éliminer les résultats le moins pertinent. Ce modèle est populaire, et plus utilisé par les moteurs de recherche actuel. (BAEZA-YATES et RIBEIRO-NETO, 1999; SOULIER, 2014; ZIANI, 2022)

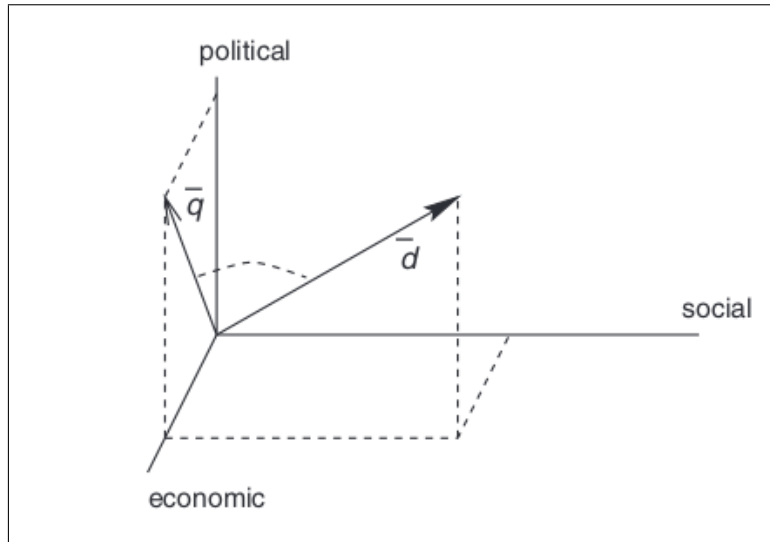


FIGURE 1.6 – Représentation du document et la requête dans le modèle vectoriel (HIEMSTRA, 1999)

Mais ce modèle a un inconvénient, qui est l'indépendance des termes d'indexation ce qui implique que la notion de sémantique du document est perdue. Mais ce problème a été solutionné par la mise place de regroupement des termes qui ont la même sens, on appelle N-grammes. Ou bien une autre approche est d'utiliser le modèle d'indexation sémantique latente (Latent Semantic Index).

Definition 1.11.3 (Modèle Vectoriel). On note w_{ij} le poids positif et non binaire, du terme i dans le document j qui est associé avec un pair (k_i, d_j) et w_{iq} le poids du terme i dans la requête q . La requête est définie par le vecteur : $\vec{q} = (w_{1q}, w_{2q}, \dots, w_{tq})$ où t le nombre total des termes d'indexation dans le système. Un document d_j est présenté par le vecteur : $\vec{d}_j = (w_{1j}, w_{2j}, \dots, w_{tj})$.

La calcul de similarité entre ces deux vecteurs se traduit par la formule :

$$\text{Sim}(\vec{d}, \vec{q}) = \frac{\vec{d}_j \cdot \vec{q}}{|\vec{d}_j| \times |\vec{q}|} = \frac{\sum_{i=1}^t w_{i,j} \times w_{i,q}}{\sqrt{\sum_{i=1}^t (w_{i,j})^2} \times \sqrt{\sum_{i=1}^t (w_{i,q})^2}}$$

Avec $|\vec{d}_j|$ et $|\vec{q}|$ norme du vecteur document et de la vecteur requête.

1.11.6.2 Autre modèle

Il y a d'autres modèles dans la catégorie algébrique tel que le modèle d'Indexation Sémantique Latente (Latent Semantic Indexing) et le modèle de réseau de neurones (Neural Network Model).

L'approche d'indexation sémantique latente est introduit en 1988, qui permet de garder la sémantique des documents pour avoir une relation entre les termes d'indexation. Cette approche se base sur l'appariement de contexte au lieu d'index, voir *Modern Information Retrieval*, Page

45 (BAEZA-YATES et RIBEIRO-NETO, 1999) pour les détails complet.

Le modèle neuronale se base sur des processus d'activation des neurones (spread activation process), cette approche est plus orienté vers l'intelligence artificielle (BAEZA-YATES et RIBEIRO-NETO, 1999).

1.11.7 Modèle probabiliste

Dans le modèle probabiliste, le framework pour modéliser les documents et la représentation de la requête se base sur la théorie des probabilités. On va pas rentrer dans les détails sur ce modèle, mais juste un tour d'horizon. Pour ce qui veulent entrée dans les détails, voir chapitre 2 de (BAEZA-YATES et RIBEIRO-NETO, 1999).

1.11.7.1 Modèle probabiliste

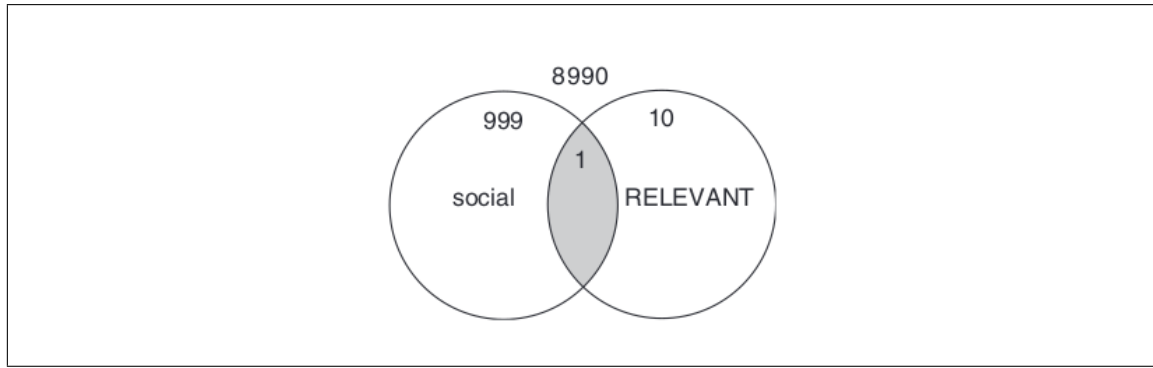


FIGURE 1.7 – Diagramme de collection pour le terme social dans le modèle probabiliste (HIEMSTRA, 1999)

Le principe du modèle probabiliste est de donner la requête q de l'utilisateur et un document d_j dans la collection. Puis le modèle estime la probabilité que l'utilisateur trouve le document d_j pertinent. Le modèle assume que ce probabilité de pertinence dépends seulement de la requête et la représentation du document. La mesure de similarité utilise la formule de Bayes (Bayes's rule). Pour plus de détails voir Section 1.4 (HIEMSTRA, 1999) ou chapitre 2 (BAEZA-YATES et RIBEIRO-NETO, 1999).

Definition 1.11.4 (Modèle probabiliste). On note $w_{ij} \in \{0, 1\}$ le poids du terme i dans le document j , et $w_{iq} \in \{0, 1\}$ le poids du terme i dans la requête q .

R : Ensemble des documents connu ou initialement estimé d'être pertinent.

\bar{R} : Complément de R , l'ensemble de documents non pertinent.

$P(R/d_j)$: Probabilité de pertinence du document d_j pour la requête q .

$P(\bar{R}/d_j)$: Probabilité de non pertinence du document d_j pour la requête q .

$$Sim(d_j, q) = \frac{P(R/\vec{d}_j)}{P(\bar{R}/\vec{d}_j)} = \frac{P(\vec{d}_j/R) \times P(R)}{P(\vec{d}_j/\bar{R}) \times P(\bar{R})}$$

Où :

$P(d_j/R)$: Probabilité de sélection du document d_j au hasard provenant de l'ensemble R des documents pertinents.

$P(R)$: Probabilité qu'un document sélectionné au hasard dans la collection entière est pertinent.

Or, $P(R)$ et $P(\bar{R})$ est la même pour tous les documents, alors la formule de similarité devient :

$$Sim(d_j, q) \sim \frac{P(\vec{d}_j/R)}{P(\vec{d}_j/\bar{R})}$$

Avec l'application des poids de terme, la formule de similarité final est donc :

$$Sim(d_j, q) \sim \sum_{i=1}^t (w_{iq} \times w_{ij}) \times \left(\log \frac{P(k_i/R)}{1 - P(k_i/R)} + \log \frac{P(k_i/\bar{R})}{1 - P(k_i/\bar{R})} \right)$$

Tel qu'au départ :

$P(k_i/R) : 0.5$

$P(k_i/R) = \frac{n_i}{N}$ avec n_i le nombre de documents qui contient le terme k_i , et N le nombre total des documents.

Ce modèle a un avantage d'avoir le classement des résultats suivant leur probabilité d'être pertinent dans l'ordre décroissant. Mais a quelques inconvénients qui est le besoin de savoir initialement la séparation des documents en ensembles pertinent et non pertinent, le modèle ne prend pas en compte le fréquence de terme dans un document ce qui implique que tous les poids des termes sont binaire, ainsi les termes d'indexation sont indépendants (BAEZA-YATES et RIBEIRO-NETO, 1999).

1.11.7.2 Autres modèles

Il y a d'autres modèles dans ce catégorie, mais comme tous les autres on va pas rentrer dans les détails puisque ça va à l'encontre de notre analyse. On cite le Réseau d'Inférence (Inference Network), Réseau Bayésien (Bayesian Network), le modèle de Poisson (The 2-Poisson model) (HIEMSTRA, 1999). Pour plus de détails voir (BAEZA-YATES et RIBEIRO-NETO, 1999).

1.12 Mesure de similarité

Citons quelques méthode de similarité (ABU-SALIH, 2018; SAADOUNE, 2018; Vaibhav SINGH et Vinay SINGH, 2022) :

- **Produit scalaire (Dot Product)** : c'est le produit scalaire entre deux vecteurs. La requête et les documents sont représenté par des vecteurs de même dimension.

- **Similarité cosinus (Cosine Similarity)** : c'est la mesure la plus populaire. Utilise le produit scalaire ainsi que l'angle formé entre les deux vecteurs (document et requête)
- **Coefficient de Jaccard (Jaccard Coefficient)** : mesure statistique de similarité entre des collections (sample set)
- **Coefficient de Dice (Dice Coefficient)** : même que celle de Jaccard, mais double le poids de vecteur

Puisque nous-nous intéressons au modèle vectoriel dans le cadre de ce mémoire, on va analyser les mesures de similarité utilisé dans ce modèle. Il y a principalement quatre mesures de similarité comme on a présenté dans la section 1.10.9. Ces mesures sont utilisé pour déterminer la pertinence système c'est à dire la pertinence des documents par rapport a la requête de l'utilisateur. A bien noter que tous les poids des termes d'indexation sont normalisé avec TF-IDF avant la calcul de ces mesures. Ces mesures sont optionnelles, mais la mesure de similarité le plus populaire est celle de Cosinus (Vaibhav SINGH et Vinay SINGH, 2022).

La pondération des termes se traduit alors comme suit :

$$d_k = q_k = tf(k, d) \cdot \log \frac{N}{df(k)}$$

Avec : Term Frequency (TF) :

$$TF(t, d) = \frac{\text{Nombre d'occurences du termet dans le document } d}{\text{Nombre total des termes dans le document } d}$$

Inverse Document Frequency (IDF) :

$$IDF(t, D) = \log \frac{\text{Nombre total des documents dans le corpus } N}{\text{Nombre de documents qui contient le terme } t}$$

Term Frequency-Inverse Document Frequency (TF-IDF) :

$$TF - IDF(t, d, D) = TF(t, d) \times IDF(t, D)$$

1.12.1 Produit scalaire

Le produit scalaire ou Dot product (Inner Product) en anglais est la mesure de base, qui est utilisé dans d'autres mesure comme celle de cosinus, Jaccard ainsi que Dice. Cette mesure calcul le produit scalaire entre le vecteur document et la vecteur requête. La formule pour calculer ce mesure est ci-dessous :

$$InnerProduct(\mathbf{A}, \mathbf{B}) = \mathbf{A} \cdot \mathbf{B}$$

1.12.2 Similarité cosinus

La mesure cosinus ou Cosine Similarity est la mesure la plus populaire. Cette mesure utilise le produit scalaire en introduisant l'angle formé par les deux vecteur (document et requête). La formule pour calculer cette mesure est ci-dessous :

$$Sim(\vec{d}, \vec{q}) = \frac{\vec{d} \cdot \vec{q}}{\|\vec{d}\| \cdot \|\vec{q}\|} = \frac{\sum_{k=1}^n d_k \cdot q_k}{\sqrt{\sum_{k=1}^n (d_k)^2} \cdot \sqrt{\sum_{k=1}^n (q_k)^2}}$$

1.12.3 Similarité de Jaccard

La mesure de Jaccard ou Jaccard's Similarity utilise aussi le produit scalaire, c'est une mesure basé sur la statistique. La formule pour calculer cette mesure est ci-dessous :

$$Jaccard\ Similarity(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

1.12.4 Similarité de Dice

La mesure de Dice ou Dice's Similarity est la mesure utilise le produit scalaire. La formule pour calculer cette mesure est ci-dessous :

$$Dice\ Similarity(A, B) = \frac{2|A \cap B|}{|A| + |B|}$$

1.13 Conclusion

En guise de conclusion de cette partie qui concerne la recherche d'information, qu'il y a beaucoup des chercheurs qui s'y intéressent puisque c'est un domaine vaste, intéressant mais aussi difficile et possède des défis. C'est un domaine ancien mais qui a connu des évolutions considérable jusqu'à nos jours avec l'explosion de volume d'information disponible sur internet par le biais des réseau sociaux, des blogs, etc. On a vue dans cette partie qu'il y a des modèles mathématique appliqués dans ce modèle ce qui rend la recherche formalisé. Ainsi l'amélioration de ces modèles par le biais de l'intelligence artificielle est possible, ce qui est le cas des géants des moteurs de recherche modernes aujourd'hui.

On a vue dans cette partie qu'il y a beaucoup des modèles utilisés dans la RI, des mesures et d'algorithme de calcul de similarité ainsi que de classement des résultats.

Analysons de plus près dans la section suivante ce qui concerne le Système de Recherche d'Information (moteur de recherche).

Chapitre 2

Système de recherche d'information

2.1 Introduction

Dans cette section, on va voir d'abord quelques définitions tirées des travaux de quelques chercheurs qui s'y intéressaient dans le domaine de la Recherche d'information et d'une encyclopédie, puis on va analyser ses composants et les types, et enfin on va voir comment analyser la performance d'un moteur de recherche et les mesures de performance souvent utilisées.

Definition 2.1.1. Selon SALTON, un système de recherche d'information (RI) est un système qui permet de retrouver les documents pertinents à une requête d'utilisateur, à partir d'une base de documents volumineuse (SALTON, 1989).

Definition 2.1.2. On appelle Système de Recherche d'Information (SRI) un ensemble des programmes, qui sert à interfacer avec l'utilisateur, de prendre les requêtes et de les interpréter afin de faire une recherche dans l'index, appliquer un modèle d'appariement et retourner les documents jugés pertinents à l'utilisateur (MAZARI, 2022).

Definition 2.1.3. On appelle SRI ou moteur de recherche, un ensemble de programmes informatiques, qui met en œuvre des techniques et moyens pour trouver les documents pertinents afin de satisfaire le besoin d'information de l'utilisateur. En d'autres termes, permet de retrouver des informations grâce à l'utilisation des mots clés et termes de recherche (ZIANI, 2022).

Definition 2.1.4. Un Système de Recherche d'Information permet de récupérer la requête de l'utilisateur, de l'analyser, puis rechercher dans la base documentaire les documents qui correspondent (pertinents) à la requête et de retourner les résultats à l'utilisateur (Vaibhav SINGH et Vinay SINGH, 2022).

Definition 2.1.5. Un moteur de recherche est un outil qui permet de rechercher sur le Web ou sur un ordinateur personnel des ressources, des contenus, des documents etc., à partir de mots

clés (JOURNAL DU NET, 2023).

Definition 2.1.6. Un moteur de recherche est un système logiciel qui collecte des informations à partir des ressources documentaires (World Wide Web, ...) et qui les présente aux utilisateurs qui recherchent des informations spécifiques (MOZILLA DEVELOPER NETWORK, 2023).

2.2 Objectif et efficacité

En général il y a deux catégorie de problème tel que le problème *centré utilisateur* qui est le problème lié a la compréhension et l'analyse de comportement de l'utilisateur ; et le problème *centré système* qui est le problème lié a la création d'index efficace, traitements des requêtes avec plus de rapidité et efficacité ainsi que le développement d'un algorithme de classement pour améliorer les résultats (BAEZA-YATES et RIBEIRO-NETO, 1999). Mais le problème central d'un SRI se concentre sur le coté système (*centré système*) qui est de savoir comment satisfaire la besoin d'information de l'utilisateur le plus rapide possible et efficace. C'est un problème plus délicat a résoudre puisqu'il y en a plusieurs facteurs relié a ces performances. Mais il y a des recherches qui traite (analyse) les comportement de l'utilisateur ainsi que ses impacts sur la recherche d'information et l'utilisation d'un SRI. Parmi ces différentes études, il y a celle qui est traité par KIEN QUACHTAT (KIEN QUACHTAT, 2012) dans le cadre des lycéens.

L'objectif d'un SRI est donc de localiser et retourner les documents pertinents par rapport a une requête pour répondre et satisfaire le besoin d'information de l'utilisateur. Le SRI relève donc le défi de rechercher des documents pertinents par rapport au requête de l'utilisateur parmi un très grand volume d'informations (MAZARI, 2022).

L'efficacité d'un SRI est jugé par la capacité de comprendre ce que l'utilisateur veut, ce que l'utilisateur demande a partir de la requête. Ainsi sur sa capacité de fournir les documents pertinents et les classer suivant leurs pertinences par rapport a la requête de l'utilisateur (Vaibhav SINGH et Vinay SINGH, 2022). Et aussi la rapidité de récupération des informations et l'accessibilité au niveau de l'interface utilisateur (YANG, 2000).

Pour notre cas, on se focalisera plus sur la performance relié au problème centré système que ceux qui est centré utilisateur.

2.3 Élément principaux

Un Système de Recherche d'Information a systématiquement deux éléments principaux, tel que la *base de donnée* et l'*algorithme de classement* (ZIANI, 2022). Ces deux éléments sont indispensables pour un système d'information.

2.3.1 Base de données

Definition 2.3.1. Une base de données est une collection organisée d'informations structurées, généralement stockées électroniquement dans un système informatique. Généralement contrôlée par un SGBD (Système de Gestion de Base de Données) (ORACLE, 2023).

Definition 2.3.2. Une base de données ou BDD est une collection d'informations organisées afin d'être facilement consultables, gérables et mises à jour. Les bases de données informatiques sont utilisées dans un grand nombre d'entreprises pour stocker, organiser et analyser les données (LEBIGDATA, 2023).

Definition 2.3.3. Une base de données permet de stocker et de retrouver des données structurées, semi-structurées ou des données brutes ou de l'information, souvent en rapport avec un thème ou une activité (WIKIPEDIA, 2023).

Pour un SRI, cette base de données est structuré par un expert en base de données (Database Manager). Il est responsable de stockage des documents, des index, ainsi que met en place la relation entre les documents et les index (Fichier inversé). Cette base de données stocke alors l'ensemble de documents utilisé par le système afin de répondre aux besoins de l'utilisateur (ABU-SALIH, 2018).

2.3.2 Algorithme de classement

Un algorithme de classement permet de classer les documents suivant l'ordre de pertinence par rapport à la requête de l'utilisateur. Cette algorithme qui se charge de calculer la pertinence entre la requête et les documents dans la base de données. L'algorithme de classement souvent utilisé dans le modèle vectoriel est déjà présenté dans la section 1.12.

2.4 Catégorie de recherche

Tout d'abord, il y a quatre catégories de recherche cité par YANG (YANG, 2000) tel que :

- **Recherche simple** : utilise la requête simple
- **Recherche personnalisée** : utilise la requête personnalisée
- **Recherche de dossier** : recherche dans de dossier
- **Recherche des nouvelles courantes** : utilise la requête des news
- **Contenu web** : suppression des liens mortes

2.5 Type de moteur de recherche

Les moteurs de recherche sont catégorisé en deux catégories : moteur de recherche classique qui travaille avec les contenues statique des documents et le moteur de recherche sémantique (BELLAOUAR, DJOUDI et ZIDAT, 2009 ; ZIANI, 2022).

Ci-dessous un tableau qui illustre la différence entre un moteur de recherche classique et moderne (YANG, 2000) (A analyser si ce tableau est vraiment nécessaire).

2.5.1 Classique ou traditionnel

Un moteur de recherche classique ou traditionnel traite les documents dont le contenu est statique. Le contenu des documents ne change pas au fil du temps. Ce qui veut dire que l'étape d'indexation des documents se fait une fois, il n'y a probablement pas de réindexation des documents. Cette approche est souvent utilisé pour la recherche des livres électronique, etc.

2.5.2 Sémantique ou moderne

Ce moteur de recherche provient du traditionnel. Le contenu des documents traités par ce type de SRI sont dynamiques et peuvent varier au fil du temps. Comme par exemple un site WEB qui met a jour fréquemment ses contenus, modifié ou supprimé. Dans de type de SRI, l'étape d'indexation est de plus en plus complexe, et l'étape de réindexation est nécessaire et souvent periodique pour récupérer les mise a jours des documents. Ce type de moteur de recherche est généralement utilisé pour la recherche sur le WEB.

2.6 Modèle adopté par un SRI

Un Système de Recherche d'Information peut adopter un modèle qui se base sur ces trois modèles qui est le modèle *full-text*, le modèle *keyword-based* et le modèle *hypertext* (BAEZA-YATES et RIBEIRO-NETO, 1999).

- **Full-text** : ce modèle utilise tous les termes de documents comme un terme d'indexation. C'est à dire pas de discrimination des mots vides. L'indexation et la tache de recherche sont simple, mais peut apporter des bruits.
- **Keyword-based** : ce modèle utilise les mots clés pour représenter les documents, avec la suppression des mots vides. Les termes d'indexation sont donc des mots clés.
- **Hypertext** : ce modèle se base sur des liens, souvent utilisé sur les documents WEB.

2.7 Évaluation d'un SRI

Dans cette section, on va analyser comment bien définir la performance et évaluer un SRI. Il est important d'évaluer un SRI avant d'implémentation pour l'utilisation finale. Cette étape d'évaluation dépend du but finale du SRI. Généralement, il y a quatre trois catégories d'évaluation d'un SRI : *évaluation fonctionnel* qui fait l'analyse des fonctionnalités du système, *évaluation de performance* et l'*évaluation de performance de recherche* (BAEZA-YATES et RIBEIRO-NETO, 1999).

Dans ce troisième catégorie, on utilise des mesures mathématique dont les plus courant sont *Précision (Precision)* et *Rappel (Recall)*.

Souvent dans l'étape d'évaluation, l'*évaluation de performance de recherche* est plus difficile et compliqué, qui a un grand défi.

2.7.1 Évaluation fonctionnel

L'évaluation fonctionnel est la première étape d'évaluation d'un SRI. Il met en œuvre l'analyse des deux facteurs suivantes :

- **Analyse de fonctionnalités** : analyser tous les fonctionnalités du système une par une pour voir s'il y a bien des failles ou bug. Cette analyse assurera que l'utilisateur finale du système faire face a des problèmes de fonctionnalités ou bug (analyse de l'interface graphique si c'est conviviale, l'accessibilité, etc.).
- **Analyse d'erreur** : pour l'analyse d'erreur, on cherche a trouver un moyen pour faire échouer le système, trouver par tous les moyens de faire des traitements pour que le système fait un erreur. Cette analyse permet de voir des erreurs qui ne sont pas identifié avant et de pouvoir les corriger après.

2.7.2 Évaluation de performance

L'évaluation de performance permet d'identifier la performance et la robustesse d'un SRI. Cette évaluation analyse généralement deux facteurs :

- **Analyse de temps de réponse** : analyser le temps de réponse du système pour répondre a une requête de l'utilisateur. Le but est de minimiser ce temps de réponse pour satisfaire l'utilisateur. Le système doit être rapide en terme de réponse. Avec les machines de plus en plus performants, cet analyse est de moins en moins problématique mais a ne pas ignorer.
- **Analyse de l'espace utilisé** : analyser aussi l'espace disque utiliser par le stockage des index et la collection des documents. Le système doit utiliser le moins d'espace possible, et doit être le plus léger possible.

Dans cette évaluation, on analyse aussi la performance d'indexation du SRI, l'interaction avec le Système d'Exploitation, ainsi que le délais dans le canaux de communication.

2.7.3 Évaluation de performance de recherche

L'évaluation de performance de recherche possède un grand défi, est qui est généralement traité par des chercheurs. Le but est de maximiser la précision des résultats de recherche par rapport au requête de l'utilisateur. Pour bien assimiler cette partie, on va voir comment est structuré la collection des documents du point de vue du système, on va illustrer ce qu'on appelle matrice de contingence dans la Figure 2.1.

| | | |
|------------|---------------------------|-------------------------------|
| irrelevant | Sélection. & Non Pert. | Non sélection. & Non Pert. |
| | Sélection. & Pert | not sélection. mais Pert. |
| | retrieved | not retrieved |

FIGURE 2.1 – Matrice de contingence (ABU-SALIH, 2018)

Dans ce matrice, il y a quatre catégories tel que :

- **Récupéré et non pertinents** : se sont des documents non pertinents mais jugés pertinent par le système. On cherche a minimiser le nombre des documents dans cette catégorie. On les appelle faux positif.
- **Récupéré et pertinents** : se sont des documents pertinents et qui sont sélectionnés par le système. On cherche a maximiser le nombre des documents dans cette catégorie. On l'appelle vrai positif.
- **Non récupéré et non pertinents** : se sont des documents non pertinents qui ne sont pas sélectionnés par le système. On cherche aussi a maximiser le nombre des documents dans cette catégorie. On l'appelle vrai négatif.
- **Non récupéré et pertinents** : se sont des documents pertinents mais qui ne sont pas récupérer par le système. On cherche a minimiser le nombre des documents dans cette catégorie. On l'appelle faux négatif.

Pour cette évaluation, on va voir quelques mesures parmi les plus populaires, tel que la *précision*, le *rappel*, et la *F-mesure*. A bien noter que la notion d'ensemble des documents pertinents dans cette évaluation est défini par un expert, qui juge qu'un document est pertinent par rapport a un requête. C'est grâce au connaissance de documents pertinents a l'avance qu'on peut juger le système.

On va illustrer dans la Figure 2.2 la classe des documents pour mieux élaborer la notion de précision et rappel.

2.7.3.1 Précision

Le précision (precision en anglais) est la fraction du nombre de documents pertinent par le nombre de documents retournés par le système. En d'autre terme, la fraction des documents retournés qui sont pertinents. La précision calcule alors la performance du système a récupérer seulement les documents pertinents (ABU-SALIH, 2018 ; BLAIR et MARON, 1985 ; YANG, 2000).

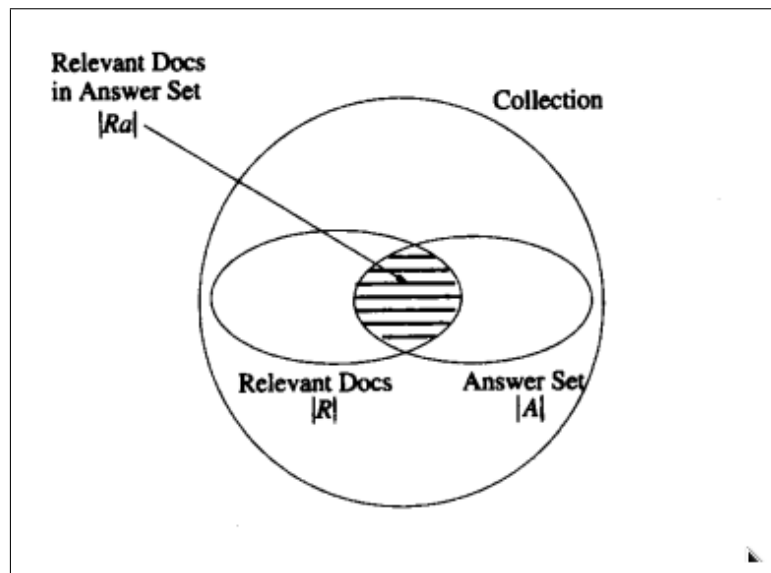


FIGURE 2.2 – Ensemble des documents (BAEZA-YATES et RIBEIRO-NETO, 1999)

Le but est de maximiser cette valeur.

$$Precision = \frac{\text{Nombre de documents pertinents}}{\text{Nombre de documents retournés par le système}} = \frac{|Ra|}{|A|}$$

2.7.3.2 Rappel

Le rappel (recall en anglais) est la fraction entre le nombre de documents pertinents retournés divisé et le nombre total des documents pertinents. En d'autre terme, c'est la fraction des documents pertinents qui sont retournés par le système. Le rappel permet de savoir comment le système a bien récupérer tous les documents pertinents (ABU-SALIH, 2018 ; BLAIR et MARON, 1985 ; YANG, 2000). Le but est de maximiser cette valeur.

$$Precision = \frac{\text{Nombre de documents pertinents}}{\text{Nombre de tous les documents pertinents}} = \frac{|Ra|}{|R|}$$

Il est nécessaire de noter que le rappel n'est efficace dans le cas où la base de données devient large, car le rappel diminue si la base de données augmente.

2.7.4 Exemple de précision et rappel

Un exemple de calcul de précision et rappel est illustré dans la Figure 2.3.

Et la courbe (relation) entre la précision et le rappel est illustré dans la Figure 2.4

2.7.5 Problème de précision et le rappel

(A discuter et a compléter - Modern IR)

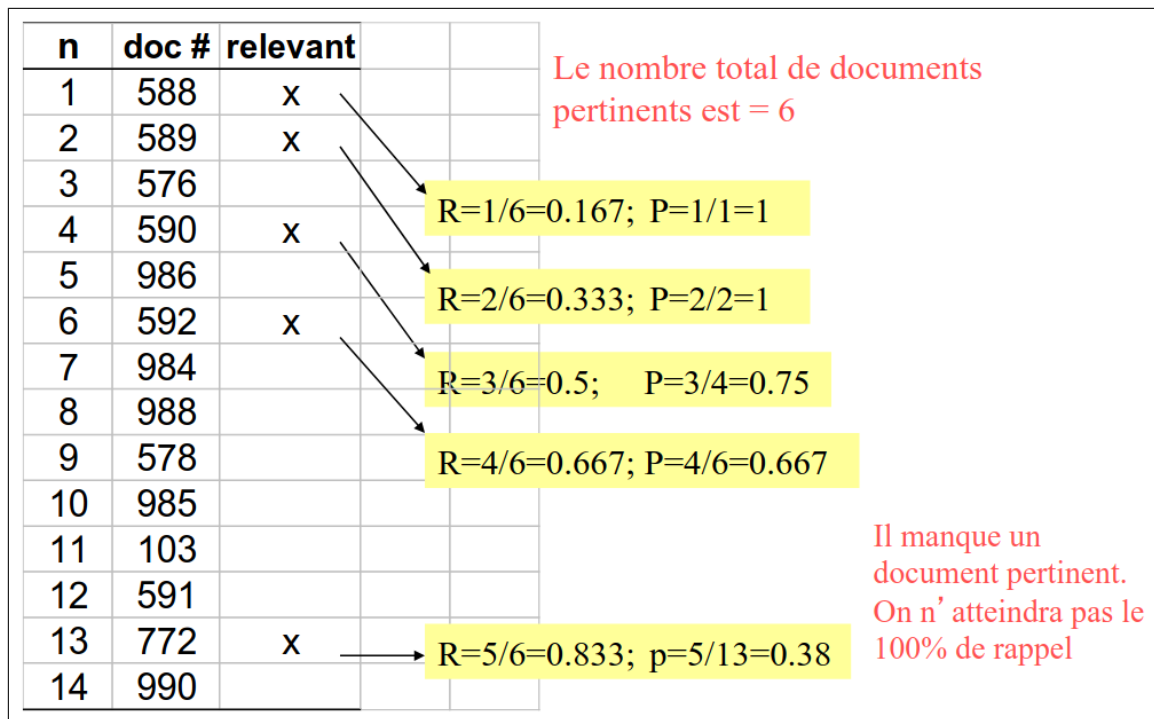


FIGURE 2.3 – Exemple de calcul de précision et rappel (BOUGHANEM, s. d.)

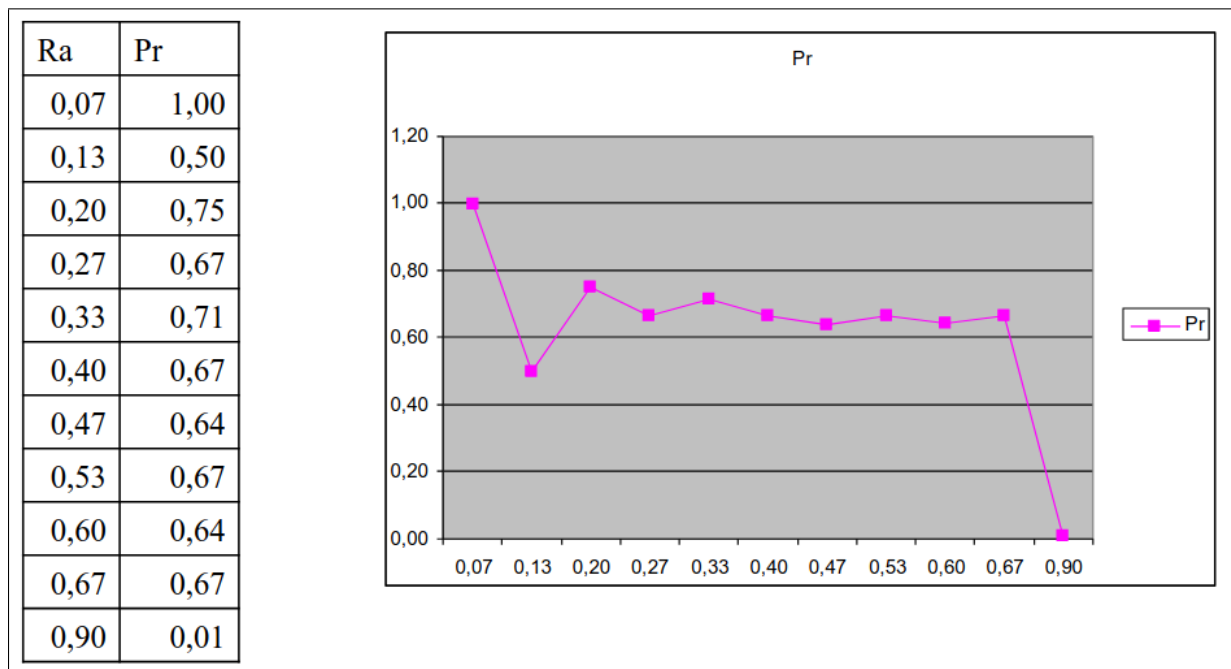


FIGURE 2.4 – Courbe précision-rappel (BOUGHANEM, s. d.)

2.7.6 Moyenne harmonique

Cette mesure utilise la précision et le rappel.

$$F(j) = \frac{2}{\frac{1}{R(j)} + \frac{1}{P(j)}}$$

Où :

$R(j)$ est le rappel pour le j -ième document.

$P(j)$ est la précision pour le j -ième document.

$F(j)$ est la moyenne harmonique

2.7.7 Et d'autres mesures

D'autres mesure de performance sont utilisés comme *accuracy*, *F-measure*, *E-measure*, *x-precision*, etc. Il y a aussi de notion de silence et bruit (BAEZA-YATES et RIBEIRO-NETO, 1999 ; MAZARI, 2022).

2.8 Analyse du moteur de recherche existant

Il y a beaucoup des moteurs de recherche où on peut trouver des thèses malagasy mais dans le cadre de ce mémoire, on va analyser quelques uns qui sont populaire et qui concerne le mieux notre étude.

2.8.1 Thèses malgache en ligne

Thèse malgache en ligne (UNIVERSITÉ D'ANTANANARIVO, 2016) est un SRI orienté WEB spécialisé pour la recherche des thèses soutenues dans les six universités publiques de Madagascar depuis 2006. Actuellement il regroupe actuellement **31160** documents, plus précisément des thèses. Ce SRI est hébergé dans le serveur de l'université d'Antananarivo, et est mis en place par les équipes au sein de l'université. Le statistique des documents est dans la figure 2.5, les résultats de recherche ainsi que la pagination est illustré dans la figure 2.6 et 2.7.

Ce SRI a pour objectif de faciliter les recherches de thèses, généralement d'auteur malagasy afin d'explorer et de mettre en évidence les fruits de recherche malagasy. Ainsi de faciliter l'accès a ces documents pour les enseignants ainsi que pour les étudiants qui cherche des travaux rattachés a son thème.

Ce SRI a un avantage majeur, qui est la rapidité de la recherche. Aussi il y a la possibilité de rechercher par un titre, auteur et contenu, ainsi que la mise en place de système de pagination pour classer les documents plus pertinents dans la première page. Ce SRI classe les résultats suivant l'ordre de pertinence décroissant. L'interface utilisateur est très simple ce qui implique



FIGURE 2.5 – Statistique des documents (UNIVERSITÉ D'ANTANANARIVO, 2016)

que le chargement de la page est rapide.

Ce Système a quelques lacunes malgré ses avantages. La première problème qui peut être remédié est l'insuffisance de nombre des documents stockés, car vu le nombre d'étudiants ayant soutenu dans les six universités publiques de Madagascar, c'est pas suffisant. La deuxième, les documents sont limités pour les établissements publiques, alors qu'il pourra être intéressant d'inclure les documents des établissement privés. La troisième c'est qu'il y a pas de système de catégorisation des documents. Et d'autres problèmes comme le nombre de documents retournés sont nombreux qui implique souvent beaucoup de pagination des résultats (pas de limite de pertinence), il est impossible de voir la résumé d'un document particulier ou afficher une aperçu du document, et finalement il y a pas de notion de sécurité de document c'est a dire tous les documents sont exposés publiquement en tant qu'utilisateur anonyme.

2.8.2 Bibliothèques universitaire en ligne

Les bibliothèques universitaire en ligne n'est plus ni moins qu'une version numérique des bibliothèques. Chaque université publique de Madagascar possède un. En général, c'est pas un SRI complet mais juste une liste des documents disponibles avec un simple recherche généralement sur le titre ou auteur.

L'objectif est d'exposer des documents pour faciliter l'accès aux étudiants, enseignants ou des personnes voulant accéder a des ressources. Mais en général, ce système fonctionne très bien avec un recherche simple dans le cas des livres par exemple, mas s'avère compliqué pour des recherches avancés. Ainsi les documents sont incomplètes.

Dans le cadre de recherche des thèses malagasy, ce système est moins efficace que Thèse malgache en ligne.

2.8.3 Google Scholar, theses.fr et d'autres

Les géants du moteurs de recherche académique tel que google scholar, Mémoire online, HAL, et d'autres peuvent bien être utilisé pour rechercher des thèses malagasy. Ces systèmes sont performants, puissant en terme d'algorithme utilisés, et stock une grande volume des documents. Certains système propose d'aperçu du document, exportation des bibliographie, liste des articles connexes ainsi qu'un système de sécurité et d'authentification. Il est aussi possible de faire une filtre par année ou par catégorie.

Malgré tous ces atouts, c'est pas suffisant pour les travaux de recherches malagasy. Ces moteurs n'arrivent pas a indexer beaucoup des documents (thèses) malagasy. Par exemple, dans thèses.fr, il y a seulement 41 thèses en cotutelle pour l'université d'Antananarivo (THESES.FR, 2023). Alors pour rechercher des travaux de recherche malagasy dans ces moteurs est un véritable défi.

2.9 Conclusion

En résumé, un Système de Recherche d'Information ou Moteur de Recherche est un système actuellement indissociable de la vie quotidienne, académique, personnel et professionnel. Il a pour but de satisfaire le besoin d'information de l'utilisateur en utilisant des algorithmes et des traitements particulières sur les documents et la requêtes. Un SRI doit passer certains test avant d'être finalement implémenté pour l'utilisation finale. Ces testes sera réalisé pas des experts du domaine, et en utilisant des corpus de test (collection de test) pour pouvoir déterminer la pertinence et la performance du SRI. Il y a actuellement beaucoup des moteurs de recherche, certains sont spécialisés dans un domaine spécifique, mais on a analysé justement quelques un qui est important dans le cadre de ce devoir.

Chapitre 3

Traitement de Langage Naturel (TLN)

3.1 Introduction

Le Traitement de Langage Naturel (Natural Language Processing) est une approche pour analyse des textes qui est basé sur les théories et la technologies. C'est un domaine très répandu actuellement dans la recherche et développement (LIDDY, 2001). La NLP est utilisé dans la recherche d'information, traduction des textes, traduction des voix, système de question réponse ainsi que diverses domaines.

Definition 3.1.1 (NLP). Natural Language Processing (NLP) est un discipline qui combine la linguistique, l'informatique et l'intelligence artificielle pour analyser et étudier les interactions entre uns système informatique et une langage naturel humain. La Recherche d'Information est parmi l'une de tâche du NLP (ANDREA FERRARIO, 2020).

Definition 3.1.2 (NLP). Selon LIDDY, un NLP est un ensemble des techniques informatiques théoriques motivés pour analyser et représenter naturellement des textes (un langage quel-conque) en passant par un ou plusieurs niveaux d'analyse linguistique dans le but de parvenir a traiter le langage humain dans un étendu de tâches ou applications (LIDDY, 2001).

L'analyse de langage, porte a connaître la structure de langage tel que les mots, les significations, combinaison des mots, ainsi que la contribution des mots au sens de la phrase. Et aussi d'analyser le fonctionnement du monde et le raisonnement de l'humanité dans le monde (TANNIER, 2006).

Le domaine de TLN a quelques divisions tel que :

- **NLP** : production de représentation significatif. Équivalent du rôle de lecteur et écouteur.
- **NLG** : production de langage a partir d'une représentation. Équivalent du rôle d'auteur et interlocuteur.
- **Language Understanding** : Termine avec un langage orale.
- **Speech Understanding** : Démarre avec un langage orale. Permet de savoir comment

un langage orale va se traduire en texte.

Le domaine de TLN par rapport a l'intelligence artificielle est illustré dans la Figure 3.1.

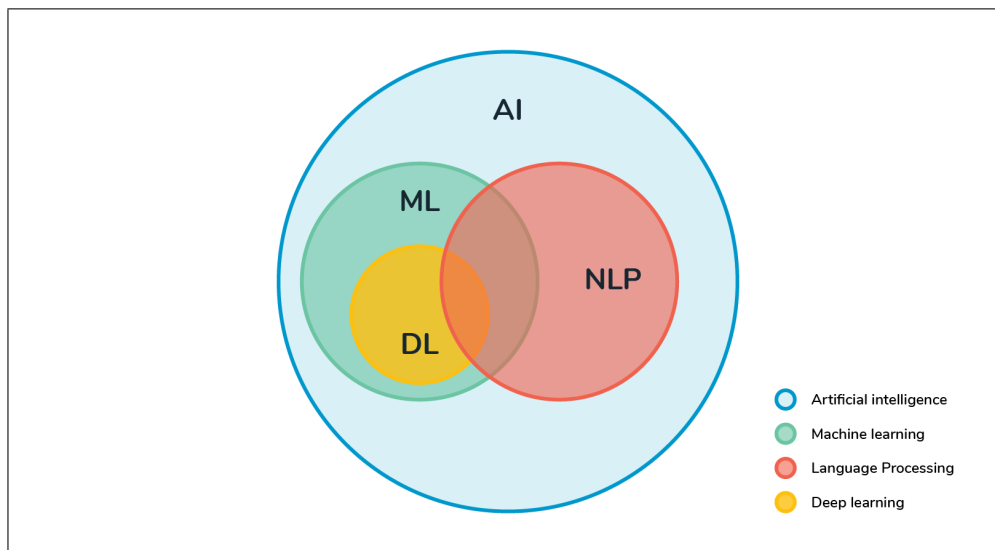


FIGURE 3.1 – Position du TLN par rapport a l'IA (DEVOPEDIA, Year)

Certains notion qu'on abordera ici sont déjà présenté dans la section 1.10, donc certains notions ne seront pas détaillés.

3.2 Un peu d'histoire

Cette brève historique est tiré de l'ouvrage de LIDDY (LIDDY, 2001). La recherche en Traitement de Langage Naturel trouve ses origines dans les années 1940. Le MT (Machine Translation) est la première application en NLP qui est une application d'ordinateur. En 1946, Weaver et Booth démarre l'une de projet récent de MT, encore sur la traduction basé sur l'expertise de voler le code d'ennemie durant la deuxième guerre mondiale. Ce projet a suggérer l'utilisation de la cryptographie et de la théorie de l'information pour le traduction de langage. Puis la recherche sont devenu varié dans les institutions en États-Unis d'Amérique (USA).

En 1957, Chomsky a publié *Syntactic Structure*, qui introduit l'idée de grammaire générative pour remédier aux problèmes rencontré dans les années antérieurs tel que l'ambiguïté syntaxique, ... Une autre domaine de NLP commence aussi a émerger, comme la reconnaissance vocale.

La communauté de Traitement de langage et la communauté vocale (Speech community) était divisé alors en deux camps : d'une part la communauté de traitement de langage dominé par la théorie perspective de grammaire germinative et hostile en méthode statistique ; et d'autre part la communauté vocal (Speech Community) dominé par la théorie statistique de l'information et hostile en théorie linguistique.

En 1950, l'homme pense que la meilleur qualité totalement automatique peut produire des

résultats indiscernable de la translation humaine, et qu'un tel système peut être opérationnel dans quelques années.

Enfin, la NLP a connu beaucoup d'évolution ainsi des nombreuses lacunes. A partir des années 1980, la NLP a évolué beaucoup plus vite par la dispositions des ressources computationnel ainsi que des ordinateurs performants.

3.3 Les niveaux de langage

Un langage comporte généralement cinq niveaux ou étape de traitement (DALE, ROBERT ET MOISL, HERMANN ET SOMERS, HAROLD, s. d. ; TANNIER, 2006), illustré dans la Figure 3.2 :

- **Phonétique et phonologie** : la liaison des mots et des phrases au son qui les réalisent à l'oral. Cette niveau n'est pas utile dans le traitement de langage textuelle.
- **Morphologie** : la façon dont les mots sont construits, et de connaître leurs rôles dans la phrase.
- **Syntaxe** : la façon dont les mots se combinent pour former des *syntagmes*, puis des proposition et enfin des phrases correctes. Un **syntagmes** est un unité syntaxique intermédiaire entre le mot et la phrase, et comprenant un seul noyau et souvent des compléments.
- **Sémantique** : la façon dont les mots font du sens lorsqu'ils sont insérés dans une phrase indépendamment du contexte.
- **Pragmatique** : la façon d'interpréter les phrases selon leur contexte d'énonciation comme l'interlocuteur, phrase précédente, connaissance commune du monde, ...

3.4 Phonologie

La phonologie interprète le langage parlé (son de parole) dans un mot (LIDDY, 2001). Il y a trois règles tel que :

- **Règle phonétique** : pour les sons dans les mots.
- **Règle phonémique** : pour les variations de prononciation quand les mots son prononcé ensemble.
- **Règle prosodique** : pour la variation d'un stresse et d'intonation dans un phrase.

3.5 Morphologie

La morphologie est l'étude de la forme des mots tel que leur *flexion* : indication de cas, genre, nombre, mode et temps ; leur *dérivation* : préfixes, suffixes et infixes ; leur *composition* : mots composés. C'est aussi une analyse morphosyntaxique, qui analyse des règles de combinaison des morphèmes selon la configuration syntaxique de l'énoncé. La morphologie consiste à segmenter des textes en unités élémentaire qu'on appelle *tokenisation* et à déterminer les différentes caractéristiques de ces unités (LIDDY, 2001 ; TANNIER, 2006).

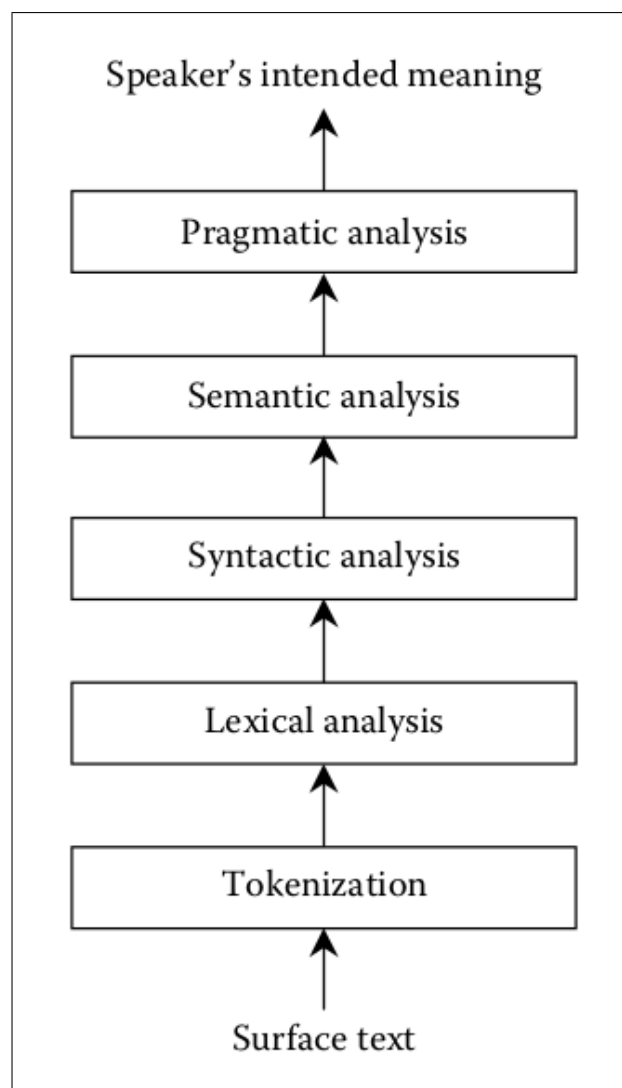


FIGURE 3.2 – Étape d'analyse de texte dans NLP (DALE, ROBERT ET MOISL, HERMANN ET SOMERS, HAROLD, s. d.)

Un humain est capable de faire ce traitement pour comprendre le sens, vu que chaque sens des morphèmes ne change pas dans tous les mots, et idem pour un système NLP qui consiste à reconnaître le sens porté par chaque morphème pour représenter le sens (LIDDY, 2001).

Definition 3.5.1 (Lemme). On appelle *lemme* la racine d'un mot, sans ses marques d'accord, de conjugaison, de cas. L'entrée d'un dictionnaire est généralement un lemme. Le lemme est nécessaire dans toute analyse sémantique. (TANNIER, 2006)

Definition 3.5.2 (Flexion). Les *flexions* sont les modifications opérées sur le lemme pour distinguer les formes de conjugaison (personne, temps, mode, voix – flexion verbale) ou le genre, le nombre et le cas (flexion nominale). L'opération qui consiste à retrouver ces informations est la *lemmatisation*, ou souvent appelé *racinisation* (*stemming*) (TANNIER, 2006).

Definition 3.5.3 (Morphème). On appelle *morphème* le plus petit unité ou unité minimale de sens (LIDDY, 2001). Par exemple, le mot prétraitement est composé de trois morphèmes tel que le préfixe **pre**, la racine **traite** et la suffixe **ment**.

Cette étape est une partie essentiel du NLP, qui est nécessaire pour définir les caractères, mots et phrases dans un document. Cette relève un défi tel que la résolution des ambiguïtés dans le langage naturel ainsi que de convertir des fichiers textes dans un séquence de texte bien défini avec un sens. Cette étape peut être divisé en deux parties : *Triage de document* (*Document Triage*) et *Segmentation de texte* (*Text Segmentation*) (DALE, ROBERT ET MOISL, HERMANN ET SOMERS, HAROLD, s. d.).

3.5.1 Triage de document

Le triage de document c'est l'action de convertir un ensemble des fichiers en documents textuelles bien définie. Il se décompose en trois étapes tel que :

1. **Identification d'encodage de caractère (Character Encoding Identifier)** : permet d'identifier l'encodage de caractère utilisé dans le fichier à convertir, par exemple si le fichier utilise l'encodage UTF-8 ou ISO ou autres.
2. **Identification de langage (Language Identification)** : permet de déterminer dans quelle langue le fichier est écrit (Français, Anglais, etc.).
3. **Sectionnement de texte (Text Sectionning)** : permet d'identifier le contenu textuel du fichier pour pouvoir le convertir en document textuel bien défini.

3.5.2 Segmentation de texte

La segmentation de texte est une étape cruciale et complexe dans la Traitement de Langage Naturel. Il consiste à convertir un document textuel bien définie en composant des mots et des phrases (DALE, ROBERT ET MOISL, HERMANN ET SOMERS, HAROLD, s. d.). C'est aussi la transformation d'un alignement d'un caractère en une unité élémentaire souvent des mots et phrase (TANNIER, 2006). Généralement, il y a trois étapes tel que la *segmentation de mot*

(*Word segmentation*), la *normalisation de texte* (*Text Normalization*) et la *segmentation de phrase* (*Sentence Segmentation*).

Pour la segmentation de texte, il faut définir une liste de caractères délimiteur pour pouvoir segmenter le texte (espace, ponctuation, apostrophe). Pour les langages dont l'espace n'est pas un délimiteur comme la langue chinoise par exemple, cette segmentation est plus complexe (DALE, ROBERT ET MOISL, HERMANN ET SOMERS, HAROLD, s. d.). Et même avec la langue qui a comme délimiteur les trois cités ci-dessus, il est encore compliqué de séparer les mots, car l'apostrophe peut être utilisé dans un seul mot comme *aujourd'hui*, aussi une unité élémentaire peut contenir des espaces comme *pomme de terre* (TANNIER, 2006).

En pratique le système de segmentation du texte utilise une liste de séparateurs par défaut, à laquelle ils ajoutent des connaissances lexicales et morphosyntaxiques pour traiter les cas ambigus. Chaque langue possède ses propres connaissances lexicales.

Segmentation de mot

La segmentation de mot c'est l'action couper une séquence des caractères en texte par la localisation de délimiteur de mot (*Word boundaries*) pour pouvoir être utilisé dans le traitement linguistique. Un délimiteur permet de définir le point où un mot est terminé et qu'un autre commence. Les mots obtenues sont appelées des *tokens*, et la façon d'obtenir ces tokens est la *tokenisation ou tokenization en anglais*. (DALE, ROBERT ET MOISL, HERMANN ET SOMERS, HAROLD, s. d.)

Pour les langages délimités par un espace (*space-delimited*), souvent des langues européennes, les mots sont séparés par des espaces. Tandis que pour des langues qui ne sont pas délimités par un espace comme le Chinois et le Thaï, il n'y a pas d'indication sur le délimiteur (DALE, ROBERT ET MOISL, HERMANN ET SOMERS, HAROLD, s. d.). Dans le cadre de ce devoir, on travaillera sur des langues qui ont comme délimiteur l'espace.

Dans une langue délimitée par un espace, les ponctuations sont souvent traitées comme des tokens séparés, mais qui varient d'une langue à une autre. Mais souvent pose des problèmes comme des abréviations, les guillemets et les apostrophes. (DALE, ROBERT ET MOISL, HERMANN ET SOMERS, HAROLD, s. d.)

Normalisation de texte

La normalisation de texte permet de fusionner les différentes formes de token en une forme canonique normalisée par exemple *Mr* et *Monsieur*. Cette normalisation permet aussi de normaliser des dates, des heures, des formats monétaires et d'autres informations numériques. Par exemple, les trois tokens écrits *\$300* peuvent être prononcés comme « trois cent dollars » et la normalisation de texte peut convertir le token original en token désiré. (DALE, ROBERT ET MOISL, HERMANN ET SOMERS, HAROLD, s. d.)

Segmentation de phrase

C'est de déterminer comment le texte doit être divisé en phrase pour un traitement plus avancé. En identifiant les délimiteurs de phrase (souvent un point pour la langue délimité par un espace). Souvent la segmentation de phrase est référencé comme *détection de délimiteur de phrase* ou *désambiguïsation de délimiteur de phrase* ou encore *reconnaissance de délimiteur de phrase* (DALE, ROBERT ET MOISL, HERMANN ET SOMERS, HAROLD, s. d.).

3.5.3 Catégories grammaticales

Les catégories grammaticales est une information morphosyntaxique importante. Ces catégories sont souvent des noms, verbe, adjectif, mais qui varie selon le système. Par exemple *Penn Treebank* contient 45 catégories. Sur la Figure 3.3 un exemple de catégorie grammaticale pour la phrase « *Qui veut noyer son chien l'accuse de la rage* », et qui illustre aussi la problème d'ambiguïté pour cette phrase. En général 25% du lexique est de forme ambigu pour la langue française. Et un exemple de description morphosyntaxique illustré dans la Figure 3.4.

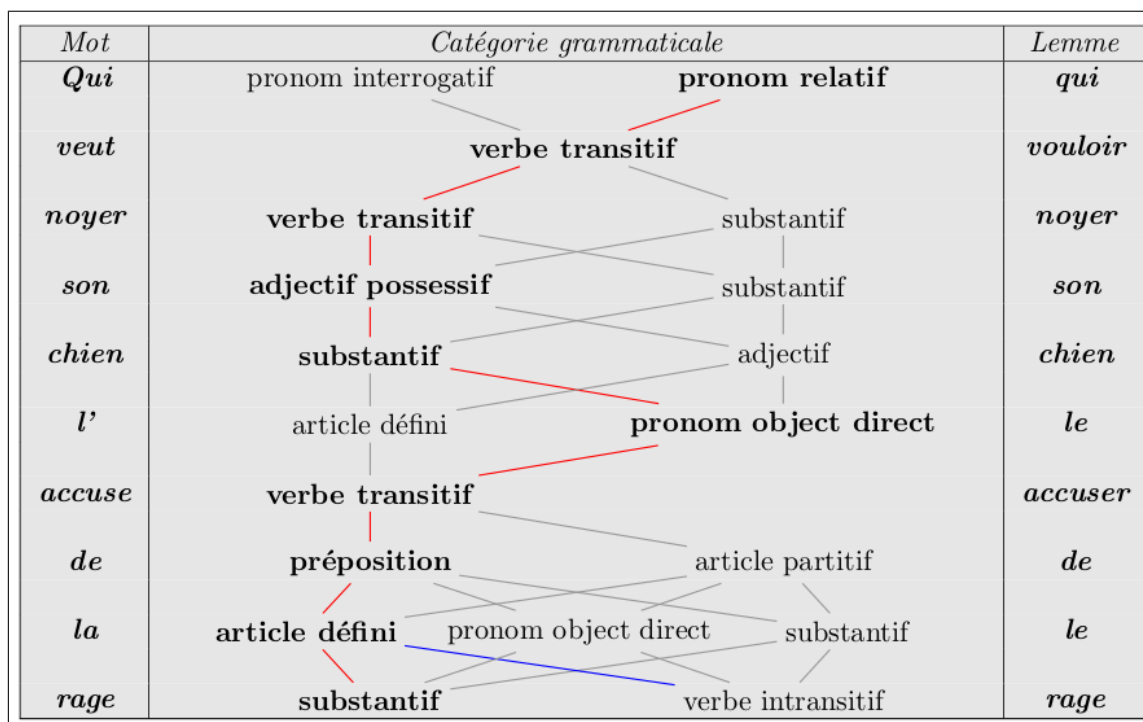


FIGURE 3.3 – Catégorie grammaticale (DALE, ROBERT ET MOISL, HERMANN ET SOMERS, HAROLD, s. d.)

3.6 Analyse lexicale

Un humain ou un Système NLP interprète le sens de chaque mot individuellement. L'analyse lexicale permet alors de déterminer le sens de chaque mot individuellement. Le but est de contribuer au compréhension de niveau de mot par une affectation de tag *part-of-speech* pour chaque mot. Un mot qui a un sens possible peut être remplacé par un représentation sémantique de ce sens (LIDDY, 2001).

| <i>veut</i> | <i>rage</i> |
|---|--|
| <div> <div>cat : verbe</div> <div>type : transitif</div> <div>lemme : vouloir</div> <div>mode : indicatif</div> <div>temps : présent</div> <div>personne : 3s</div> <div>voix : active</div> </div> | <div> <div>cat : nom</div> <div>lemme : rage</div> <div>genre : féminin</div> <div>nombre : singulier</div> </div> |

FIGURE 3.4 – Description morphosyntaxique (DALE, ROBERT ET MOISL, HERMANN ET SOMERS, HAROLD, s. d.)

Un analyse lexicale utilise un connexion, qui est déterminé et choisi par le système NLP. Un lexicon peut être **simple** c'est à dire des mots et ses *pos* ou peut être **complexe** et contient l'information de classe sémantique des mots (LIDDY, 2001).

3.7 Syntaxe

La syntaxe décrit comment les lemmes ou flexions sont ordonnées pour créer des constituants, composant aux-mêmes des phrases. Souvent, l'analyse syntaxique est représenté de façon hiérarchique. Dans l'analyse syntaxique, on travaille avec les mots, les phrases, ainsi que l'étape de la formation du mots a la phrase (TANNIER, 2006). L'analyse syntaxique est une analyse complexe avec différentes niveaux, les détails ne seront pas traité dans ce devoir, mais présent dans « Traitement automatique du langage naturel pour l'extraction et la recherche d'informations », section 3 (TANNIER, 2006).

L'analyse syntaxique selon LIDDY, se focalise sur l'analyse des mots dans une phrase, pour découvrir la structure grammaticale de la phrase, qui nécessite l'utilisation de la grammaire et un parseur. Dans l'analyse syntaxique, l'ordre et le dépendance des mots contribue au sens : par exemple, la phrase « *Le chat attaque le chien* » et la phrase « *Le chien attaque le chat* » ont un sens différent car ses termes de syntaxe est différent, pourtant ils sont composé des mêmes mots (LIDDY, 2001).

3.7.1 Analyse superficielle

Le but de l'analyse superficielle ou partielle (TANNIER, 2006), est de reconnaître les syntagmes simples, non récursif d'un énoncé sans lier les uns aux autres ; d'obtenir des résultats moins riches mais plus sûr et plus rapide. Cette approche s'oppose a l'analyse profonde ou complète qui cherche a regrouper chaque phrase dans une unique représentation.

Cass est un exemple de d'analyseur partielle efficace, qui consiste en une cascade d'automates à états finis.

3.7.2 Analyse en dépendance

L'analyse syntaxique en dépendance diffère surtout de l'analyse en constituants (comme les règles hors-contexte) par le mode de représentation. Les deux approches n'ont pas de différence au niveau de leur couverture ou de leur expressivité (A reformuler). L'idée est de relier les mots et non les constituants (TANNIER, 2006).

Un exemple de grammaire de dépendance est la *Link Grammar* qui définit la lexique et les contraintes d'attachement, comme illustré dans la Figure 3.5 et la Figure 3.6

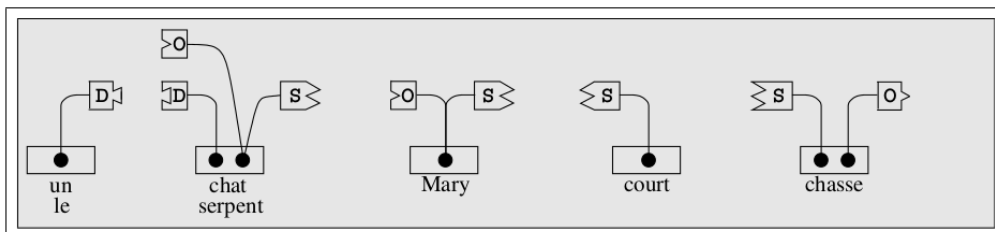


FIGURE 3.5 – Exemple de définition de la lexique *Link Grammar* (TANNIER, 2006)

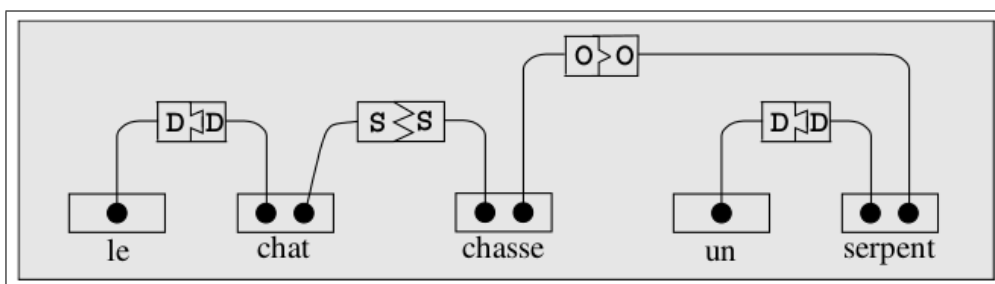


FIGURE 3.6 – Exemple d'analyse syntaxique avec *Link Grammar* (TANNIER, 2006)

3.8 Sémantique

Une approche sémantique peut être utilisée pour réduire les ambiguïtés syntaxiques, à mieux cibler des contextes (en RI par exemple), mais son but final est de représenter formellement l'information véhiculée par un énoncé et éventuellement d'en inférer de nouvelles connaissances ou une réponse à la question posée dans le cas où l'énoncé est une question (TANNIER, 2006). Il a pour but d'associer à une séquence de mots une représentation interne de son sens, tout en prenant compte l'utilisation future des résultats obtenus.

Selon beaucoup de personnes, c'est dans le niveau sémantique qu'on détermine la signification, pourtant tous les niveaux contribuent à la signification. La sémantique détermine alors la signification possible d'une phrase en se concentrant sur l'interaction parmi les significations de niveau des mots dans une phrase, avec la possibilité de désambiguïsation sémantique des mots aux multiples sens. Par exemple si on prend le mot **pile**, ça peut être une pile pour alimenter un appareil électronique, ou ça peut être une structure dans un langage de programmation informatique.

Selon TANNIER, il y a quatre méthodes :

- **L'analyse profonde** : permet d'obtenir une représentation complète de l'énoncé
- **Interprétation sémantique grammaticales** : qui s'appuie sur une analyse syntaxique totale
- **Grammaires sémantiques** : qui modélisent les données spécifiques
- **Patrons sémantique** : qui détectent les informations prédéfinies dans un texte

On distingue deux types de sémantique tel que la *sémantique grammaticale* qui se charge a construire un sens de l'énoncé globale, et la *sémantique lexical* qui étudie la participation des mots a ce sens et tous les ambiguïtés qu'ils provoquent.

MAZARI ; ZIANI (MAZARI, 2022 ; ZIANI, 2022) ont chacun développé un moteur de recherche basé sur cette notion de sémantique pour améliorer la qualité de recherche, et analyser les ambiguïtés dans un terme de recherche des utilisateurs.

Quelques relations importantes entre les mots

Il y a des relations importantes entre les mots (TANNIER, 2006) qui est nécessaire en analyse sémantique tel que :

- **Polysémie et l'homonymie** : propriété de certains formes graphiques (signifiants) de renvoyer à plusieurs sens (signifié). Par exemple, le mot *bureau*.
- **Synonymie** : lien entre deux mots ayant la même sens.
- **Hyponymie** : relation d'inclusion entre deux mots dont l'un (hyponymie) est plus spécifique que l'autre (hyperonyme). Par exemple le mot *gorille* est un hyponymie du mot *quadrumane*, le mot *fleur* est un hyperonymie du mot *tulipe*.
- **Méronymie et holonymie** : relation de partie a tout. Par exemple, une serrure est une partie d'une cage (méronymie), un bâtiment contient une pièce (holonymie).

Ces relations existant entre les mots sont répertoriés dans des base informatiques dont le plus utilisés est **WORDNet**.

3.9 Pragmatique

LIDDY cite un niveau supplémentaire entre la sémantique et la pragmatique, c'est le **discours** qui travaille sur un texte plus longue qu'une phrase et interprète le sens en connectant tous les phrases. Il y a deux types de traitement de discours tel que : l'*Anaphore* et le *Discours/text structure recognition*.

Tandis que l'analyse pragmatique sert dans un situation bien spécifique, utilise de contexte qui dépasse le contenu de compréhension de texte (LIDDY, 2001). En d'autres termes, regroupe un grand nombre de domaine qui englobent tous les problèmes qui ne pouvant être traités avec la syntaxe et la sémantique (TANNIER, 2006). Cette analyse nécessite l'utilisation des

connaissances extra-linguistiques sur le contexte et du discours. Pour cette raison que les applications pratiques sont rares, pourtant il y a quelques uns (TANNIER, 2006) comme la *déictique*, l'*implicateurs conversationnelle* et la *présupposition*.

3.10 Approche du TLN

Il y a généralement trois approches dans le Traitement de Langage Naturel, tel que l'*approche symbolique*, l'*approche statistique* et l'*approche connexionniste*.

3.10.1 Approche symbolique

Cette approche a coexisté avec l'approche statistique le jour après la naissance de ce domaine. Elle fait une analyse profonde des phénomènes linguistiques, et qui est basé sur la représentation explicite des faits à propos d'un langage à travers une connaissance bien comprise. Cette approche a été trouvée dans un système basé sur des règles (ensemble des règles, moteur d'inférence et un espace de travail) ; ou logique : structure sans forme de proposition logique.

L'approche symbolique est appliquée dans divers domaines de recherche comme l'*extraction d'information*, la *catégorisation de texte*, *résolution d'ambiguïté*, et l'*acquisition lexicale*.

3.10.2 Approche statistique

Cette approche, souvent utilisée un large volume de texte (corpora), emploi des techniques mathématiques variées pour développer des modèles approximatifs généralisés de phénomène linguistique. Cette approche utilise des données observables comme source primaire d'évidence. Le modèle statistique souvent utilisé est le *HMM* ou *Hidden Markov Model*.

L'approche statistique est appliquée dans la *reconnaissance vocale*, *acquisition lexicale*, *parsing*, *part-of-speech tagging*, *machine de translation statistique*, *collocations*, *apprentissage statistique de grammaire*.

3.10.3 Approche connexionniste

Cette approche est apparue dans les années 1960, est comme l'approche statistique qui développe des modèles généralisés depuis l'exemple de phénomène linguistique. Elle combine l'apprentissage statistique avec différentes *théories de représentation* qui permet la transformation, l'inférence et la manipulation de formule logique.

Certains modèles de l'approche connexionniste s'appellent **modèle localiste** qui peut faire des tâches comme *désambiguïsation de sens de mot*, *génération de langage*, et *inférence limitée* ; et **modèle distribué** qui est utilisé pour les tâches comme la *parseur des syntaxes* (*syntactic parsing*), *translation dans un domaine limité ou spécifique* et *recherche associative*.

3.11 Application de TLN

La Traitement de Langage Naturel s'applique dans diverses domaines de recherche que ce soit textuelle ou sonores.

La Recherche d'Information (RI)

Dans la recherche d'information, on travaille sur des textes, ce qui implique que certains implémentation utilise le NLP. Cette application utilise souvent l'approche statistique (LIDDY, 2001). Le TLN est généralement utilisé dans le niveau traitement morphologique.

Elle analyse la variation morphologique des mots ou *stemming*, mais aussi la variation syntaxique (étude des syntagmes nominaux) ainsi que la variation sémantiques : relation liant des lemmes différents mais identiquement proches et la multitude de sens qui peut prendre une forme graphique donnée (polysémie, homonymie) (TANNIER, 2006).

Stemming

(TANNIER, 2006) La plus courante utilise une approximation des phénomènes linguistiques d'une langue donnée, comme les mécanismes habituels de conjugaison, d'accord ou de genre et en nombre, ou sa dérivation et tente de supprimer les suffixes tout en regroupant les différentes allomorphes (variante graphique d'une même racine).

L'algorithme le plus utilisé est celle de *Lovins* et *Porter* pour la langue anglaise et *Jacques Savoy* pour la langue française.

L'Extraction d'Information (EI)

Ce domaine d'application est plus récente, qui se base sur la reconnaissance, marquage (tagging), et extraction dans un représentation structurés, certains éléments clés d'information, par exemple : personne, compagnie, localisation, organisation, d'un large collection de texte (TANNIER, 2006).

Et d'autres applications

Et d'autres domaines d'applications, comme la machine de traduction (Machine Translation), système de dialogue, système de question-réponse, système de résumé (summarization).

3.12 Conclusion

En conclusion, le Traitement de Langage Naturel est l'un de domaine intéressant, et contribue a beaucoup des recherches dans la domaine de l'informatique. Il est aussi nécessaire dans la recherche d'information. Elle se divise en deux principale branches qui est le traitement de

langage écrite et le traitement de langage orale. Elle propose différentes approches et méthodes ainsi que différentes algorithmes pour faire des traitements. Elle est utilisés dans différentes domaines et qui est devenue indissociable de l'intelligence artificielle.

Chapitre 4

Modèle vectoriel

4.1 Introduction

Le modèle vectoriel (Vector Space Model) est l'une de modèle le plus utilisé en recherche d'information, que ce soit textuel (SAADOUNE, 2018) ou multimédia (MARTINET, 2004). Ce modèle est apparu lorsque la pondération binaire est limitant, et que l'appariement partielle n'est pas possible, en introduisant la pondération non binaire et un appariement partielle (BAEZA-YATES et RIBEIRO-NETO, 1999).

Le principe de ce modèle vectoriel modélise la requête et des documents sous forme d'un vecteur, pondérer les termes dans ce vecteur avec des méthodes de pondération comme le *TF-IDF*, et calcule ensuite la distance euclidienne entre les vecteurs documents et la vecteur requête pour déterminer la pertinence (YANG, 2000). Un score de similarité est alors calculé pour pouvoir classer les documents jugés pertinent par le système. La représentation du document et la requête est illustré dans la Figure 1.6 dans la section 1.11.6.1.

Definition 4.1.1 (Modèle Vécotriel). On note w_{ij} le poids positif et non binaire, du terme i dans le document j qui est associé avec un pair (k_i, d_j) et w_{iq} le poids du terme i dans la requête q . La requête est définie par le vecteur : $\vec{q} = (w_{1q}, w_{2q}, \dots, w_{tq})$ où t le nombre total des termes d'indexation dans le système. Un document d_j est présenté par le vecteur : $\vec{d}_j = (w_{1j}, w_{2j}, \dots, w_{tj})$.

La calcul de similarité entre ces deux vecteurs se traduit par la formule :

$$Sim(\vec{d}, \vec{q}) = \frac{\vec{d}_j \cdot \vec{q}}{|\vec{d}_j| \times |\vec{q}|} = \frac{\sum_{i=1}^t w_{i,j} \times w_{i,q}}{\sqrt{\sum_{i=1}^t (w_{i,j})^2} \times \sqrt{\sum_{i=1}^t (w_{i,q})^2}}$$

Avec $|\vec{d}_j|$ et $|\vec{q}|$ norme du vecteur document et de la vecteur requête.

Dans le cadre de ce mémoire, c'est cette modèle qu'on utilisera, et on l'appliquera sur des

documents PDF ou WORD.

4.2 Problème de classification

Ce problème est illustré et analysé dans (BAEZA-YATES et RIBEIRO-NETO, 1999). L'étude de Salton définit la problème de RI comme un problème de classification. Notons une collection de document C et que la requête de l'utilisateur est une vague représentation de l'ensemble des documents A . Le problème est donc de déterminer quelle documents appartient à l'ensemble A , qui est un problème de classification.

Dans un problème de classification, il y a deux problèmes : la première consiste à déterminer les fonctionnalités qui décrivent le mieux les objets dans la collection A ; et la seconde consiste à déterminer les fonctionnalités qui distinguent les objets dans la collection A pour les autres objets provenant de la collection C . La première fonctionnalité produit la quantification *intra-clustering* tandis que la seconde fonctionnalité produit la quantification *inter-clustering*.

Dans le modèle vectoriel, la similarité *intra-clustering* est quantifiée par la mesure de la fréquence du terme k_i dans le document d_j . Ce facteur est souvent appelé facteur *tf* ou *term frequency*, qui décrit la caractérisation intra-document. D'autre part, la similarité *inter-clustering* est quantifiée par la mesure de l'inverse de la fréquence du terme k_i parmi les documents dans la collection. Ce facteur est souvent appelé facteur *idf* ou *inverse document frequency*.

Pour avoir une meilleure classification des documents, il faut balancer ces deux mesures.

4.3 Méthodes de pondération des termes

Cette mesure permet de calculer la fréquence d'un terme dans un document 1.12.

Definition 4.3.1. Cette définition est tirée de *Modern Information Retrieval* (BAEZA-YATES et RIBEIRO-NETO, 1999). Notons N le nombre total des documents dans le système et n_i le nombre de documents contenant le terme k_i . Notons $freq_{i,j}$ la fréquence du terme k_i dans le document d_j (le nombre de fois où le terme k_i est mentionné dans le texte du document d_j). Alors la fréquence normalisée $f_{i,j}$ du terme k_i dans le document d_j est donnée par

$$f_{i,j} = \frac{freq_{i,j}}{\max_l freq_{i,l}}$$

où le maximum est calculé à travers les termes qui sont mentionnés dans le texte du document d_j . Si le terme k_i n'appartient pas au document d_j alors $f_{i,j} = 0$. D'autre part, notons *idf_i* l'*inverse document frequency* pour le terme k_i , qui est donnée par

$$idf_i = \log \frac{N}{n_i}$$

La pondération de terme le plus connu et plus utilisé est donné par

$$w_{i,j} = f_{i,j} \times \log \frac{N}{n_i}$$

ou par la variation de ce formule. Ce pondération de termes est aussi appelé facteur *tf-idf*

Certains variations de ce methode de pondération est introduit en 1988, mais la formule précédente est efficace pour la plupart des collection. Pour la requête, Salton et Buckley (BAEZA-YATES et RIBEIRO-NETO, 1999) a suggéré la pondération suivante

$$w_{i,q} = \left(0.5 + \frac{0.5 \text{freq}_{i,q}}{\max_l \text{freq}_{i,j}} \right) \times \log \frac{N}{n_i}$$

4.4 Variant du TF-IDF

Les variants de ces méthodes sont illustré dans la Figure 4.1 qui est celle de la *tf*, la Figure 4.2 qui est celle de l'*idf* et la Figure 4.3 qui est celle de *tf-idf*.

| weighting scheme | tf weight |
|--------------------------|--|
| binary | 0, 1 |
| raw count | $f_{t,d}$ |
| term frequency | $f_{t,d} / \sum_{t' \in d} f_{t',d}$ |
| log normalization | $\log(1 + f_{t,d})$ |
| double normalization 0.5 | $0.5 + 0.5 \cdot \frac{f_{t,d}}{\max_{\{t' \in d\}} f_{t',d}}$ |
| double normalization K | $K + (1 - K) \frac{f_{t,d}}{\max_{\{t' \in d\}} f_{t',d}}$ |

FIGURE 4.1 – Variant du TF (SAADOUNE, 2018)

4.5 Document inversé

Le document inversé ou fichier inversé (Inverted index) permet de représenter les documents ainsi que l'ensemble des termes dans la base de données, ainsi que ses poids calculés par ce méthode de pondération.

| weighting scheme | idf weight ($n_t = \{d \in D : t \in d\} $) |
|--|--|
| unary | 1 |
| inverse document frequency | $\log \frac{N}{n_t} = -\log \frac{n_t}{N}$ |
| inverse document frequency smooth | $\log \left(\frac{N}{1 + n_t} \right) + 1$ |
| inverse document frequency max | $\log \left(\frac{\max_{t' \in d} n_{t'}}{1 + n_t} \right)$ |
| probabilistic inverse document frequency | $\log \frac{N - n_t}{n_t}$ |

FIGURE 4.2 – Variant de l'IDF (SAADOUNE, 2018)

| weighting scheme | idf weight ($n_t = \{d \in D : t \in d\} $) |
|--|--|
| unary | 1 |
| inverse document frequency | $\log \frac{N}{n_t} = -\log \frac{n_t}{N}$ |
| inverse document frequency smooth | $\log \left(\frac{N}{1 + n_t} \right) + 1$ |
| inverse document frequency max | $\log \left(\frac{\max_{t' \in d} n_{t'}}{1 + n_t} \right)$ |
| probabilistic inverse document frequency | $\log \frac{N - n_t}{n_t}$ |

FIGURE 4.3 – Variant du TF-IDF (SAADOUNE, 2018)

4.6 Avantages

Le modèle vectoriel a certains avantages : Facile a mettre en œuvre, possibilité d’avoir un appariement approximatif avec un degré de pertinence (L’utilisateur reçoit des documents qui pourrait lui intéressé), possibilité d’organiser les résultats suivant leur pertinence (L’utilisateur passe moins de temps a filtrer les résultats puisqu’ils sont déjà ordonné), ainsi ce modèle peut définir une limite pour la mesure de similarité et de n’afficher que les documents qui sont en dessus de cette limite pour éliminer les résultats le moins pertinent. Ce modèle est populaire, et plus utilisé par les moteurs de recherche actuel. (BAEZA-YATES et RIBEIRO-NETO, 1999; SOULIER, 2014; ZIANI, 2022)

4.7 Inconvénients

Mais ce modèle a un inconvénients, qui est l’indépendance des termes d’indexation ce qui implique que la notion de sémantique du document est perdue. Mais ce problème a été solutionné par la mise place de regroupement des termes qui ont la même sens, on appelle N-grammes. Ou bien une autre approche est d’utiliser le modèle d’indexation sémantique latente (Latent Semantic Index).

4.8 Conclusion

Conclusion sur le VSM

Chapitre 5

Simulation et Réalisation

5.1 Approche utilisée

5.1.1 UML

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

5.2 Technologies utilisé

5.2.1 Python

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy

eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

5.2.2 MySQL

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

5.3 Outils utilisée

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras

viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

5.3.1 Visual Studio Code

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Conclusion générale

Nulla in ipsum. Praesent eros nulla, congue vitae, euismod ut, commodo a, wisi. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Aenean nonummy magna non leo. Sed felis erat, ullamcorper in, dictum non, ultricies ut, lectus. Proin vel arcu a odio lobortis euismod. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Proin ut est. Aliquam odio. Pellentesque massa turpis, cursus eu, euismod nec, tempor congue, nulla. Duis viverra gravida mauris. Cras tincidunt. Curabitur eros ligula, varius ut, pulvinar in, cursus faucibus, augue.

Nulla mattis luctus nulla. Duis commodo velit at leo. Aliquam vulputate magna et leo. Nam vestibulum ullamcorper leo. Vestibulum condimentum rutrum mauris. Donec id mauris. Morbi molestie justo et pede. Vivamus eget turpis sed nisl cursus tempor. Curabitur mollis sapien condimentum nunc. In wisi nisl, malesuada at, dignissim sit amet, lobortis in, odio. Aenean consequat arcu a ante. Pellentesque porta elit sit amet orci. Etiam at turpis nec elit ultricies imperdiet. Nulla facilisi. In hac habitasse platea dictumst. Suspendisse viverra aliquam risus. Nullam pede justo, molestie nonummy, scelerisque eu, facilisis vel, arcu.

Curabitur tellus magna, porttitor a, commodo a, commodo in, tortor. Donec interdum. Praesent scelerisque. Maecenas posuere sodales odio. Vivamus metus lacus, varius quis, imperdiet quis, rhoncus a, turpis. Etiam ligula arcu, elementum a, venenatis quis, sollicitudin sed, metus. Donec nunc pede, tincidunt in, venenatis vitae, faucibus vel, nibh. Pellentesque wisi. Nullam malesuada. Morbi ut tellus ut pede tincidunt porta. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam congue neque id dolor.

Donec et nisl at wisi luctus bibendum. Nam interdum tellus ac libero. Sed sem justo, laoreet vitae, fringilla at, adipiscing ut, nibh. Maecenas non sem quis tortor eleifend fermentum. Etiam id tortor ac mauris porta vulputate. Integer porta neque vitae massa. Maecenas tempus libero a libero posuere dictum. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aenean quis mauris sed elit commodo placerat. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Vivamus rhoncus tincidunt libero. Etiam elementum pretium justo. Vivamus est. Morbi a tellus eget pede tristique commodo. Nulla nisl. Vestibulum sed nisl eu sapien cursus rutrum.

Nulla non mauris vitae wisi posuere convallis. Sed eu nulla nec eros scelerisque pharetra.

Nullam varius. Etiam dignissim elementum metus. Vestibulum faucibus, metus sit amet mattis rhoncus, sapien dui laoreet odio, nec ultricies nibh augue a enim. Fusce in ligula. Quisque at magna et nulla commodo consequat. Proin accumsan imperdiet sem. Nunc porta. Donec feugiat mi at justo. Phasellus facilisis ipsum quis ante. In ac elit eget ipsum pharetra faucibus. Maecenas viverra nulla in massa.

Nulla ac nisl. Nullam urna nulla, ullamcorper in, interdum sit amet, gravida ut, risus. Aenean ac enim. In luctus. Phasellus eu quam vitae turpis viverra pellentesque. Duis feugiat felis ut enim. Phasellus pharetra, sem id porttitor sodales, magna nunc aliquet nibh, nec blandit nisl mauris at pede. Suspendisse risus risus, lobortis eget, semper at, imperdiet sit amet, quam. Quisque scelerisque dapibus nibh. Nam enim. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc ut metus. Ut metus justo, auctor at, ultrices eu, sagittis ut, purus. Aliquam aliquam.

Bibliographie

- A. ROZENKNOP** (s. d.). *Modèles en Recherche d'Information*. Cours Master Recherche Paris 13, Recherche et extraction d'information.
- ABU-SALIH**, Bilal (2018). « Applying Vector Space Model (VSM) Techniques in Information Retrieval for Arabic Language. » In : *CoRR* abs/1801.03627. URL : <http://dblp.uni-trier.de/db/journals/corr/corr1801.html#abs-1801-03627>.
- ANDREA FERRARIO**, Mara Nägelin (jan. 2020). *The Art of Natural Language Processing: Classical, Modern and Contemporary Approaches to Text Document Classification*.
- B S HARISH D S** Guru, S Manjunath (juill. 2010). *Representation and Classification of Text Documents: A Brief Review*.
- BAEZA-YATES**, Ricardo A. et Berthier A. **RIBEIRO-NETO** (1999). *Modern Information Retrieval*. ACM Press / Addison-Wesley. ISBN : 0-201-39829-X. URL : <http://www.ischool.berkeley.edu/~heerst/irbook/glossary.html>.
- BELLAOUAR**, Slimane, Mahieddine **DJOUDI** et Samir **ZIDAT** (2009). « Construction et utilisation d'un thésaurus pour la recherche d'information sur le web ». In : *Conférence Internationale des Technologies de l'Information et de la Communication, CITIC'09, Sétif, Algérie*.
- BENAYACHE**, Ahcène (2005). « Construction d'une mémoire organisationnelle de formation et évaluation dans un contexte e-learning ». Thèse de doctorat. Thèse de doct. Université de Technologie de Compiègne.
- BLAIR**, David C. et M. E. **MARON** (mars 1985). « An Evaluation of Retrieval Effectiveness for a Full-Text Document-Retrieval System ». In : *Commun. ACM* 28.3, p. 289-299. ISSN : 0001-0782. DOI : 10.1145/3166.3197. URL : <https://doi.org/10.1145/3166.3197>.
- BOUBÉE**, Nicole et André **TRICOT** (2010). *Qu'est-ce que rechercher de l'information? Nouvelle édition*. DOI: <https://doi.org/10.4000/books.pressesenssib.799>. Villeurbanne : Presses de l'enssib. ISBN : 9782375460412. URL : <http://books.openedition.org/pressesenssib/799>.
- BOUGHANEM**, Mohand (s. d.). *Evaluation des performances dans les SRI*. URL : <https://www.irit.fr/~Mohand.Boughanem/slides/RI/chap11-Evaluation.pdf>.

- DALE, ROBERT ET MOISL, HERMANN ET SOMERS, HAROLD (s. d.). *Handbook for Natural Language Processing*.
- DEVOPEDIA (Year). *Natural Language Processing*. Accessed on 2023-09-08. URL : <https://devopedia.org/natural-language-processing>.
- DR. LOUNNAS BILAL (2023). *Information retrieval (IR)*. Coours de recherche d'information, Université Mohamed Boudiaf - M'Sila.
- HIEMSTRA, Djoerd (1999). *Model de recherche d'information*.
- JÉRÔME VELO (2009). « Interactions entre un utilisateur et un systeme informatique : application a un systeme d'e-administration ». Thèse de doctorat en cotutelle pour l'obtention du grade de docteur en informatique des universites de rouen et d'antananarivo. Thèse de doct. Université d'Antananarivo.
- JOURNAL DU NET (2023). *Moteur de recherche : définition, traduction et acteurs*. URL : <https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203539-moteur-de-recherche-definition-traduction-et-acteurs/>.
- KIEN QUACHTAT (2012). « Recherche d'information sur le WEB et moteurs de recherche: le cas des lycéens ». Thèse de doctorat. Ecole normal supérieur de CACHAN.
- LEBIGDATA (2023). *Base de données*. URL : <https://www.lebigdata.fr/base-de-donnees>.
- LIDDY, Elizabeth D (2001). « Natural language processing ». In.
- MARTINET, Jean (déc. 2004). « Un modèle vectoriel relationnel de recherche d'information adapté aux images ». Theses. Université Joseph-Fourier - Grenoble I.
- MAZARI, Ahmed Cherif (2022). « Contribution à l'amélioration de la recherche d'information par utilisation des méthodes sémantiques: application à la langue arabe ». Thèse de doct. Université de Mohamed Kheider Biskra.
- MOZILLA DEVELOPER NETWORK (2023). *Search Engine*. URL : https://developer.mozilla.org/fr/docs/Glossary/Search_engine.
- ORACLE (2023). *What is Database*. URL : <https://www.oracle.com/fr/database/what-is-database/>.
- PARADIS, Francois (nov. 1996). « Un modèle d'indexation pour les documents textuels structurés ». Theses. Université Joseph-Fourier - Grenoble I. URL : <https://theses.hal.science/tel-00005009>.
- R MANIKANDAN, Dr. R Sivakumar (mars 2018). *Machine learning algorithms for text-documents classification: A review*.
- RANKS NL (2023). *French Stopwords*. Accessed: 03/08/2023. URL : <https://ranks.nl/stopwords/french>.

- SAADOUNE**, Billal (2018). « Development of search engine based on vector space model ». Mém. de mast. UNIVERSITY OF MOHAMED BOUDIAF - M'SILA. URL : <http://dspace.univ-msila.dz:8080/xmlui/handle/123456789/15946>.
- SALTON**, Gérard (1989). « Traitement automatique de texte : la transformation, l'analyse et la récupération de l'information par un ordinateur ». In : *Lecture : Addison-Wesley* 169.
- SALTON, GERARD ET WONG, ANITA ET YANG, CHUNG-SHU** (1975). « Un modèle d'espace vectoriel pour l'indexation automatique ». In : *Communications de l'ACM* 18.11. Sous la dir. de États-Unis **ACM NEW YORK NY**, p. 613-620.
- SINGH**, Vaibhav et Vinay **SINGH** (juill. 2022). *VECTOR SPACE MODEL: AN INFORMATION RETRIEVAL SYSTEM*.
- SOULIER**, Laure (2014). « Définition et évaluation de modèles de recherche d'information collaborative basés sur les compétences de domaine et les rôles des utilisateurs ». Français. ffNNT: ff. fftel-01110721f. Thèse de doct. Université de Toulouse. URL : <https://tel.archives-ouvertes.fr/tel-01110721/document>.
- TANNIER**, Xavier (2006). « Traitement automatique du langage naturel pour l'extraction et la recherche d'informations ». In : *Ecole Nationale Supérieure des Mines de Saint-Etienne* 16, p. 17.
- THESES.FR** (2023). *Theses.fr*. Accessed: 03/08/2023. URL : <https://www.theses.fr/112289355>.
- UNIVERSITÉ D'ANTANANARIVO** (2016). *Thèse malgache en ligne*. Thèses soutenu dans les 6 Universités de Madagascar depuis 2002.
- WIKIPEDIA** (2023). *Base de données*. URL : https://fr.wikipedia.org/wiki/Base_de_donn%C3%A9es.
- YANG**, Kiduk (2000). « Information Retrieval on the Web ». In : *ACM Computing Surveys*, n. pag.
- ZIANI**, Chaima (2022). « Réalisation d'un moteur de recherche avec une approche sémantique ». Mém. de mast. UNIVERSITY OF MOHAMED BOUDIAF - M'SILA. URL : <https://dspace.univ-bba.dz:443/xmlui/handle/123456789/3652>.