**Université de Blida 1 – Saad Dahlab**

**Faculté des sciences**

**Département d'Informatique**

Mémoire présenté par :

Mlles. AMROUCHE Fatma Zohra et MESSANIA Lamiss

Pour l'obtention du diplôme de Master

**Domaine :** Mathématique et Informatique

**Filière :** Informatique

**Spécialité :** Traitement Automatique de la Langue

Sujet :

# Vers une méthode d'appariement document-requête, multicritères, à base d'un Réseau de Neurones.

Soutenu le : 09 Novembre 2020, devant le jury composé de :

| | | |
|---|---|---|
| Mme. H.YEKHLEF | Université de Blida 1 | Présidente |
| Mme. Y.GHEBGHOUB | Université de Blida 1 | Examinatrice |
| Mme. M. MEZZI | Université de Blida 1 | Promotrice |

# Résumé

***''L'Information, c'est le pouvoir !''***

La Recherche d'Information (RI), l'un des premiers domaines de l'Informatique, n'a cessé d'évoluer dans le but de rationaliser le processus complexe permettant l'identification, au sein de volumes de plus en plus importants d'informations, celles qui sont potentiellement intéressantes pour l'utilisateur.

La sélection d'information pertinente et répondante aux besoins d'utilisateur se fait en passant par tout un processus de recherche, qui commence par l'application des techniques de Traitement Automatique de La Langue Naturelle (TALN), puis la pondération des termes, et fini par la récupération et le classement des documents.

A ce titre, notre travail vise à apporter des contributions sur deux axes complémentaires : d'abord l'amélioration du processus d'appariement requête-document (Mapping), puis l'amélioration du classement des documents pertinents retournés (Ranking). Nous avons obtenus les mesures de performances suivantes : 0.94, 1.00, 0.96 respectivement pour le rappel, la précision, et la F-mesure sur un sous ensemble du dataset.



**Mots-clefs :** Recherche d'Information, Système de Recherche d'Information, Réseau de Neurones, Appariement document-requête, Classement.

## *Abstract*

***''Information, is a power!''***

Information Research (IR), one of the first fields of IT, has continued to evolve in order to rationalize the complex process allowing the identification, within increasingly large volumes of information, that which is potentially of interest to the user.

The selection of relevant information that meets the user's needs is done through a whole research process, which begins with the application of Natural Language Processing (NLP) techniques. Then the weighting of terms and ended up retrieving and filing documents.

As such, our work aims to contribute on two complementary axes: first, improving the query-document matching process (Mapping), then improving the classification of the relevant documents returned (Ranking). We obtained the following performance measures: 0.94, 1.00, and 0.96 respectively for recall, precision, and F-measure on a subset of the dataset.



**Key words**: Information Retrieval, Information Retrieval System, Neural Network, Query-document Mapping, Ranking.

# ملخص

استمر استرجاع المعلومات (IR)، وهو أحد المجالات الأولى لتكنولوجيا المعلومات، في التطور بهدف تبسيط العملية المعقدة مما يسمح بتحديد المعلومات التي يحتمل أن تكون مثيرة للاهتمام للمستخدم داخل كميات متزايدة من المعلومات.

يتم اختيار المعلومات ذات الصلة التي تلبي احتياجات المستخدم من خلال عملية بحث كاملة، والتي تبدأ بتطبيق تقنيات معالجة اللغة الطبيعية (NLP)، ثم ترجيح المصطلحات، وينتهي باسترجاع وتصنيف الوثائق.

على هذا النحو، يهدف عملنا إلى تقديم مساهمات على محورين متكاملين: أولاً، تحسين عملية مطابقة المستندات -الاستعلام (التعيين)، ثم تحسين تصنيف المستندات ذات الصلة المرتجعة (الترتيب). لقد حصلنا على مقاييس الأداء التالية: 0.94، 1.00، 0.96 على التوالي للتذكر، والدقة، F -score على مجموعة فرعية من مجموعة البيانات.

**الكلمات المفتاحية**: البحث عن المعلومات، نظام البحث عن المعلومات ، الشبكة العصبية ، مطابقة طلب-مستندات ، التصنيف.

# *Dédicaces*

***A mon grand-père (Allah yerhmo) qui m'a soudainement quitté, mais qui n'a jamais quitté mes pensées…***

***A ma jolie Mamie…..A mes parents …..A mes charmantes sœurs…A ma deuxième maman 'Khalto Amina' et mes deux oncles 'Youcef et Bachir'….A ma binôme 'Lamiss'….et à moi-même je dédie ce modeste travail.***

***Fatma Zohra…***

***Je dédie ce beau travail, A mes chères parents qui ont été toujours a mes côtés et qu'ils m'ont toujours encouragé pour continuer mes études de Master.***

***A mes deux frères 'Abdelkarim' et 'Abdelkader' … A ma belle-sœur 'Wassila' pour son soutien moral et sa motivation … A 'Lynda' qui a été ma sœur avant d'être mon amie….A le meilleur binôme que l'on peut souhaiter ' ZoZo '…***

***Lamiss …***

# Remerciement

Je remercie tout d'abord chère ALLAH qui nous a donner la force et le courage de passer tous les mauvais moments ,et la volonté de faire ce travail en peu de temps .Et qui nous a prouvé à chaque fois qu'il est proche de nous et il nous a jamais laisser.

Merci papa et mama pour l'amour, pour le soutien et l'encouragement, aujourd'hui je réalise que j'ai accompli de grandes choses grâce à vos prières, a votre Douaa et vos bons conseils. Merci a toi papa car tu ma pousser pour continuer mes études de Master. Tout ce que vous faites pour moi Dieu vous remboursera Inchallah, merci beaucoup.

Mon âme sœur 'Lynda', ma source d'énergie , merci d'être toujours à mes côtés dans mes études ainsi dans ma vie personnelle, merci ma belle pour votre sincère amitié ,pour vos conseils ,ainsi les beaux mots qui m'encouragent toujours .Merci d'être avec moi dans les pires moments de ma vie, j'ai toujours pu compter sur toi.

Ma belle-sœur 'Wassila', du fond du cœur … merci pour votre amour et votre gentillesse, Tu es une personne formidable.

'Fatma Zohra' ,Ma chère binôme, je ne sais pas comment t'exprimer mes sentiments, tout d'abord je remercie Allah pour te connaitre et de nous donner l'occasion de nous réunir pour débattre ensemble dans ce travail .je n'oublierai jamais les difficultés qu'on a passé ensemble ,les nuits blanches ,les moments de dépression, de faiblesse et de désespoir , les moments de la joie , et surtout ton bon humeur et ton joli esprit . Merci infiniment, Je ne sais pas ce que j'aurais pu faire sans toi.

Ma deuxième famille, 'Tata Anissa' ,' Hadjer' , 'Ihcene' et la plus belle 'Besma' , quelle meilleure façon puis-je exprimer ma gratitude, j'ai senti que je fais partie de votre famille ,j'ai l'honneur de passer ces derniers temps chez vous , merci pour votre soutien , les mots ne suffit pas pour vous exprimer mon amour.

La meilleure pour la fin , Madame Mezzi, la douce et la gentille prof que l'on peut souhaiter , merci pour votre confiance en nous, pour vos conseils et votre soutien…Merci infiniment.

Lamiss

Je ne pourrais pas commencer le remerciement sans dire Merci dieu ,Merci de nous avoir donner la volonté , la force et la foi pour ne jamais baisser les bras et toujours continuer et avancer droit au but , الحمد لله.

Merci papa et mama , mon système de support pas seulement dans mon parcours universitaire mais durant toute mes 23 ans , ces petite lignes ne suffiraient jamais à vous remercier assez .

Merci mes trois adorables sœurs , Hadjer ma sœur aînée toi qui a été toujours là à mes coté, avec ta bonne humeur ,ton pure esprit et tes desserts délicieux ,Ihcene ma moitié, le coin de mes secrets , ma source d'énergie et d'amour éternel ,ma petite Besma celle qui viendrais chaque nuit me demander si j'ai avancer , et quand je commence à me plaindre elle m'entend avec patience même si elle ne comprenait pas tous ce que je disais , et elle finit par sourire et me dire : 'je prie pour toi Fatma' , MERCI mes sœurs parce que vous étiez et vous serez toujours ma joie et mon bonheur.

je voudrais particulièrement et chaleureusement dire MERCI à Ma Grand-Mère qui a été ma force surtout durant les 5 dernière années, et qui a vécu et vivait les pire et les bons moments durant mon parcours universitaire ,merci pour ta présence même dans les nuits blanches que je fessais , merci pour tes prières et surtout merci pour ton amour,

Je remercie également ma chère tante pour son soutien, merci ma deuxième maman.

'Ghita', Merci, pour ne pas laisser la distance nous battre, Merci d'être là , avec tes prières, ta motivation, tes appelles et ton amour ,merci d'être toi et d'être là.

'Lamiss' ! je n'ai pas pensé à la difficulté d'exprimer combien tu comptes pour moi. Tu es ma sœur qui a passé et vécu tout les up side downs dans notre parcours, je n'oublierai jamais les tard conversations qu'on avait après une longe journée de travail ...Merci pour touts nos rires, nos moments sincères ,nos moments de folies ,de joie et même de tristesse ,d'espoir et de désespoir …. Je ne trouve pas la fin entre ces lignes!! They say: 'Every journey has its own beauty ', and you were the beauty of this journey.

Enfin, je ne peux clore ces remerciements sans faire une place spéciale à notre promotrice Dr Mezzi , à mes amis et surtout 'Fetta' , à 'tata Naima' et 'Wassila' ,Merci énormément .

Fatma Zohra

Table des matières

# Liste des figures

# Liste des tableaux

## Liste des acronymes

| | |
|---|---|
| RI | Recherche d'Information |
| SRI | Système de Recherche d'Information |
| RN | Réseau de Neurone |
| SECAS | Semantically Enriched Context-Aware Stemming |
| CAS | Context-Aware Stemming |
| UNIVAC | Universal Automatic Computer |
| NIST | National Institute of Standards and Technology |
| TREC | Text Retrieval Conference |
| RSV | Retrieval Status Value |
| TF | Term Frequency |
| IDF | Inverse Document Frequency |
| LSI | Latent Semantic Indexing |
| PCC | Plus Court Chemain |
| LCS | Least Common Subsumer |
| MLP | Multi Layer Perceptron |
| CSV | Comma-Separated Values |
| NLP | Natural Langage Processing |

| | |
|---|---|
| TALN | Traitement Automatique de La Langue Naturelle |
| ANN | Artificial Neural Networks |
| CNN | Convolutional Neural Network |
| RNN | Recurent Neural Network |
| ReLu | Rectified Linear Function |
| Spyder | Scientific PYthon Development EnviRonment |

# Introduction générale

# 1 Contexte de travail

Les avancées des technologies de la communication et de stockage des données ont engendré la prolifération des quantités des données dans le web. En effet, la taille du web a été estimée à des billions de site web ces dernières années et il est encore en croissance rapide. Dans ce contexte, le problème n'est plus la disponibilité de l'information, mais la capacité de sélectionner une information qui répond aux besoins d'un utilisateur.

La recherche d'Information, l'une des plus anciennes applications de l'Informatique est la discipline qui prend soin des besoins des utilisateurs, elle s'intéresse à l'acquisition, l'organisation, le stockage et la sélection d'Informations pertinentes pour ce dernier.

Cependant les informations sont souvent inaccessibles ou difficilement accessibles. Il est, par exemple, difficile de parcourir toutes les pages retournées pour rechercher une information particulière, et c'est justement à ce stade qu'interviennent les Systèmes de Recherche d'Information (communément connus sous le nom de moteur de recherche). Parmi les tâches réalisées par ce dernier, on trouve le classement des résultats retournés (en Anglais Ranking) auquel nous allons nous intéresser dans ce projet.

# 2 Problématique et motivation

Parfois, nous devons classer les documents uniquement en fonction de leur pertinence par rapport à la requête. Dans certains autres cas, nous devons tenir compte des relations de similarités et de diversité entre les documents dans le processus de classement.

Par ailleurs, les algorithmes de Ranking en Recherche d'Information peuvent prendre du temps, et cela est dû à plusieurs raisons, comme le processus d'indexation et le calcule d'appariement requête-document qui peut se répéter lors de chaque traitement.

# 3 Objectif

Notre travail rentre dans le cadre de la continuité du travail « **Tests et analyse d'algorithmes d'indexation sémantique dans le cadre de la proposition d'un système de recherche d'information sensible au contexte** » [1], entamé l'année passée au sein du département d'Informatique de l'Université de Blida 1.

Le premier objectif de notre travail est de réaliser un système à base d'un Réseau de Neurones, afin de diminuer le temps d'exécution de la solution initiale. Plus particulièrement la tâche d'appariement document-requête, en gagnant le temps prit par le prétraitement et la ré-indexation des documents et des requêtes pour chaque opération de recherche.

Le second objectif posé par ce projet consiste à améliorer la qualité de recherche en proposant différents choix de similarité aux utilisateurs, afin d'évaluer les mesures de similarités les plus pertinentes pour une tâche de Recherche d'Information.

# 4 Organisation du mémoire

Dans le but de réaliser les objectifs mentionnés ci-dessus, notre mémoire sera devisé en deux parties principales :

- Etat de l'art, qui contient elle-même contient trois chapitres :
  - **Chapitre I** : « La Recherche d'Information ». Dans ce chapitre nous allons parler des généralités du domaine de Recherche d'Information, en passant en revues les notions les plus importantes.
  - **Chapitre II** : « Le classement en RI ». Ce chapitre est consacré au processus de Ranking, son importance, ses méthodes, les types de similarités sur lesquelles il se base, ainsi que la synthèse de deux travaux antérieurs dans le domaine.
- Conception et implémentation qui contient deux chapitres :

- **Chapitre III** : « Modélisation de la solution proposée ». Dans ce chapitre, nous parlerons de la démarche de modélisation utilisée, nous présenterons la conception du système suivant cette démarche, son architecture et ses outils de développement.
- **Chapitre IV** : « Tests et validation de la solution ». Dans ce chapitre nous évaluerons le système et la qualité des résultats qu'il fournit.

Enfin, nous conclurons le mémoire avec une conclusion qui comprend quelques perspectives à notre travail.

# Chapitre I : Recherche d’Information

# 1 Introduction

De nos jours, l'information est devenue disponible en grande quantité et en différente formats. Nous vivons dans un monde de l'information. Cette quantité énorme de données doit être accessible et contrôlable par tous les utilisateurs qui veulent y accéder. C'est ici qu'interviennent les moteurs de recherche. Et dans ce contexte, les Systèmes de Recherche d'Information(SRI) permettent de retourner des résultats qui répondent au mieux aux besoins des utilisateurs.

Ce chapitre est organisé comme suit : nous débutons avec un petit survol de l'histoire de la Recherche d'Information(RI), sa définition et quelques concepts de base de cette dernière. Par la suite nous allons parler des Système de Recherche d'Information en décrivant le processus de la RI qui inclue : l'indexation, le requêtage, l'appariement, et le classement des résultats. Après cela, nous présenterons les fonctions et les différents modèles de la RI.

# 2 Histoire de la RI

La Recherche d'Information consiste à rendre accessible les connaissances existantes. Cela n'a pas seulement été le cas depuis le début de l'ère numérique. Vannevar Bush [2] est l'un des premiers scientifiques à réfléchir sérieusement à la manière dont l'humanité peut rendre ses connaissances plus facilement accessibles face à un monde de plus en plus confus. En 1945, il a écrit l'article révolutionnaire **As We May Think**[1] dans lequel il présente une vision de l'avenir de la collecte et de l'organisation de l'information. Comme plusieurs autres domaines informatiques, les pionniers de l'époque étaient enthousiastes à utiliser l'ordinateur pour automatiser la Recherche des Informations, qui dépasse la capacité humaine car il y avait une explosion d'information après la deuxième guerre mondiale.

[1] https://www.theatlantic.com/magazine/archive/1945/07/as-we-may-think/303881/

La première description d'un ordinateur dédié à la Recherche d'Information a été décrite par Holmstrom en 1948 ; détaillant une première mention de l'ordinateur UNIVAC[2](Universal Automatic Computer) en l'an 1951. Dans les années 1960, le premier grand groupe de recherche sur la RI a été formé par Gerard Salton à Cornell, dans les années 1970, il avait été démontré que plusieurs techniques de récupération fonctionnaient bien sur de petits corpus de texte tels que la collection Cranfield[3] (plusieurs milliers de documents).

En 1992, le Département américain de la Défense et le National Institute of Standards and Technology (NIST) ont coparrainé la Text Retrieval Conference (TREC)[4] , l'objectif était d'étudier la communauté de la Recherche d'Informations en fournissant l'infrastructure nécessaire à l'évaluation des méthodologies de recherche de texte sur une très grande collection de textes.

Dans les dernières années, la recherche d'information est devenue un domaine de recherche important dans la science de l'informatique. Les systèmes de recherche d'information sont utilisés dans plusieurs domaines d'applications telles que la recherche sur le web, la recherche dans les blogs, le filtrage d'information, les agents conversationnels, et la recherche dans les réseaux sociaux, etc. [3].

# 3 Définition

D'après Salton [4], La Recherche d'Information « RI » c'est une branche de l'informatique qui s'intéresse à l'acquisition, l'organisation, le stockage et la sélection d'information pertinente pour l'utilisateur.

La RI s'intéresse au filtrage de certaines informations d'un ensemble de données répondant à des besoins utilisateurs par un programme « SRI » qui est une interface entre les utilisateurs qui expriment leurs besoins par des

[2] http://infoindustrielle.free.fr/Histoire/Fiches/Univac.htm
[3] http://ir.dcs.gla.ac.uk/resources/test_collections/cran/
[4] une série continue d' ateliers portant sur une liste de différentes recherche d'information domaines de recherche (IR)

requêtes et une grande collection de documents. Ce système sert a sélectionné les documents pertinents pour ces derniers.

# 4 Concepts et principes de base de la RI

La RI a comme principale rôle l'extraction des informations pertinentes reflétant un besoin en information, parmi un ensemble de document. De ce fait plusieurs concepts clés peuvent être définis. Nous avons donc trouvé utile de les clarifier ci-dessous.

## 4.1 Collection de documents

La collection de documents constitue l'ensemble des informations accessibles et exploitable par une entité (machine, utilisateur, etc.). Pour un souci d'optimalité, la base constitue des représentations simplifiées. Cette représentation est choisie de façon à rendre l'interrogation (recherche) et la modification (suppression ou ajout d'un document) de la base des documents faciles et rapides.

## 4.2 Besoin en information

La notion de besoin en information dans le contexte de la RI constitue le besoin des utilisateurs. Cette notion a été catégorisée par [5] en trois types :

- **Besoin vérificatif :** Dans ce cas, l'utilisateur cherche à vérifier le texte ou la description des données qu'il possède. Il recherche donc une donnée particulière, et sait même souvent comment y accédé. Ce type de besoin est stable car il ne risque pas de changer au cours de la recherche. Un exemple connu de ce besoin est la recherche de la date de publication d'un ouvrage dont la référence est connue.

- **Besoin thématique connu :** Dans une telle situation, l'utilisateur cherche à clarifier à trouver ou à avoir plus d'information dans un domaine donné. Ce type de besoin peut varier au cours du temps. En particulier, les besoins de l'utilisateur peuvent se raffiner ou s'enrichir au cours de la recherche.

- **Besoin thématique inconnu :** Dans ce cadre de besoin, l'utilisateur cherche de nouvelles informations liées à un domaine non familier. Ce besoin est essentiellement variable car les besoins des utilisateurs sont souvent incomplets vu leur ignorance du domaine de recherche.

### 4.3 Requête

C'est une interface entre l'utilisateur et le SRI, elle est constitué d'un ensemble de mots clés, elle permet d'exprimer le besoin d'information de ce dernier.

### 4.4 Modèle de représentation

Un modèle de représentation est une modélisation possible d'un document ou d'une requête, conçu d'une façon à couvrir au mieux le contenu sémantique de ces derniers. Ce processus est appelé indexation, ayant comme résultat des groupes de termes (ou de concepts). Ces termes ont différents poids et rangés dans des structures appelées dictionnaires, qui constituent les langages d'indexation.

### 4.5 Modèle de recherche

Ce modèle représente le noyau des SRI. Il permet de faire correspondre un ensemble de documents pertinents à chaque requête utilisateur.

## 5 Les systèmes de recherche d'information

Un Systèmes de Recherche d'Information (SRI) est un lien entre l'utilisateur qui exprime ses besoins avec des requêtes et la collection de document. Ce système sert à sélectionner et retourner les résultats les plus pertinents qui répondent aux besoins de ses derniers. Selon Smeaton [6] : « *Le but d'un Système de Recherche d'Information est de retrouver des documents en réponse à une requête des usagers, de manière à ce que les contenus des documents soient pertinents au besoin initial d'information de l'usager*.».

Un SRI inclut un ensemble de procédures et d'opérations qui permettent la gestion, le stockage, l'interrogation, la recherche, la sélection et la représentation de cette masse d'informations.

# 6 Le processus de la recherche d'information

Les différentes étapes du processus de RI, sont représentées schématiquement par le processus en U, illustré dans la figure suivante :

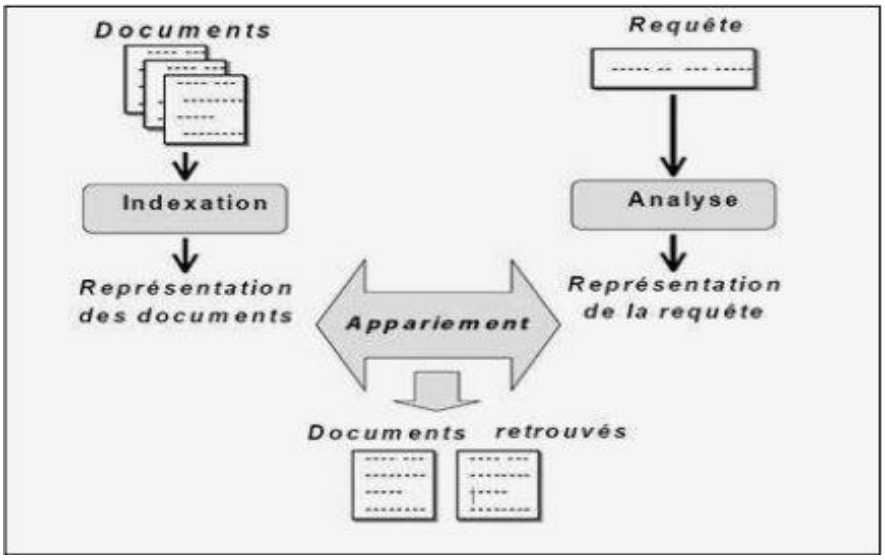



Figure 1 Représentation de processus de Recherche d'information *[7]*

Tout processus de Recherche d'Information est construit autour de 3 fonctions : l'indexation, le requêtage (recherche) et l'appariement.

## 6.1 Indexation

L'indexation est une étape très importante dans le processus de la RI. Elle a pour rôle d'extraire à partir d'un document ou d'une requête, une représentation paramétrée qui couvre au mieux son contenu sémantique en identifiant pour chaque document les termes importants, puis à exploiter ces termes comme index pour accéder rapidement aux documents. Un des

objectifs de l'indexation est donc de permettre de retrouver rapidement les documents contenant les termes (mots-clés) de la requête.

L'indexation peut être : manuelle, semi-automatique ou automatique.

**Indexation manuelle** : chaque document est analysé par un spécialiste du domaine correspondant ou par un documentaliste, qui identifie les mots clés appelés descripteurs. Elle permet la recherche par concepts (par sujets, par thèmes), et la classification de documents (par sujets, par thèmes). Cependant, l'indexation manuelle présente un effort trop coûteux en temps et en besoin humain. De plus, un degré de subjectivité lié au facteur humain fait que le même document peut être indexé de différentes façons par des personnes différentes [8].

**Indexation automatique** : est l'opération qui consiste à faire reconnaître par l'ordinateur des termes figurant dans le titre, le résumé, le texte complet, elle fait appel aux robots d'indexation, ce qui rend le processus d'indexation complètement automatisé. L'indexation automatique, basée essentiellement sur une approche statistique, est adoptée par la majorité des systèmes de RI en raison de son coût réduit par rapport à l'indexation manuelle [4] [9] [10]. Il faut noter que l'opération d'indexation automatique est difficile dans la mesure où elle pose des problèmes de l'interprétation et la représentation du sens du texte (Synonymie et Polysémie) [11].

**Indexation semi-automatique** : est basée sur un processus automatique. En outre, le choix final reste au spécialiste du domaine correspondant, qui intervient souvent pour la sélection finale des mots clés significatifs et établir des relations sémantiques entre mots-clés et choisir les termes significatifs en suivant des règles bien définies.

Généralement, l'indexation comprend une série de traitements automatisés. Ils sont appliqués sur les documents et aussi sur les requêtes. On distingue: l'extraction des mots (segmentation), l'élimination des mots vides, la normalisation et la pondération.

### 6.1.1 Extraction des mots

Cette phase consiste à segmenter le texte du document en mot, La segmentation (ou tokenization[5]) du texte est une première étape importante dans ce processus. Elle est appliquée au texte de document ainsi qu'à la requête [12].

Généralement, les « token » peuvent être des chaînes de caractères qui sont séparées par des espaces. Dans certaines langues, la tokenization est plus complexe [12]. La langue chinoise, par exemple, n'a pas de séparateur de mots clair comme un espace. Donc, une analyse lexicale est nécessaire pour identifier les "tokens" en identifiant tous ce qui peut constituer des séparateurs, des caractères spéciaux, des chiffres, des ponctuations, etc.

### 6.1.2 Élimination des mots vides

La liste des mots simples extraite précédemment peut contenir des mots non significatifs, appelés "mots vides", tels que : les pronoms personnels, les prépositions… ou même des mots athématiques qui peuvent se retrouver dans n'importe quel document (par exemple des mots comme contenir, appartenir, etc.). L'élimination de ces mots peut se faire en utilisant une liste dressée de mots vides (également appelée anti-dictionnaire), ou en écartant les mots dépassant un certain nombre d'occurrences dans la collection. Bien que ce traitement présente l'avantage de diminuer le nombre de termes d'indexation, il peut cependant induire des effets de silence (par exemple, en éliminant le mot a de vitamine a).

### 6.1.3 La normalisation

Cette phase nous permet de regrouper les différentes variantes d'un mot avec un traitement morphologique lemmatisation ou Stemming (racinisation). Tel que la lemmatisation est la transformation des variantes en lemme[6] (exemple : biologie, biologiste, biologique par : biologie) et le

[5]Tokenization : Extraction des termes tels qu'un terme est une suite de caractère séparé par (blanc, signe de ponctuation, nombre, caractères spéciaux…etc.)
[6] Lemme : Origine du mot

Stemming (racinisation ou radicalisation) concerne la transformation en stems (racine ou radical). Cela peut produire des mots qui n'ont pas de sens (exemple : économie, économiquement, économiste, économiseur va devenir : économ… un mot qui n'a pas de sens ou une entrée dans un dictionnaire). Ceci implique une perte de précision pour des requêtes telles que « économiseur de batterie ». Une solution a été proposée pour résoudre ce problème et de maximiser la proportion de stems significatives dans [13] , avec l'algorithme SECAS (Semantically Enriched Context-Awere Stemming), plus de détails sur ce dernier seront abordé dans les chapitre qui suivent.

Par ailleurs, cette phase traite aussi la capitalisation (NEuRone → neurone) et supprime les accents et les signes diacritiques (apparaît → apparait).

### 6.1.4 Pondération des termes

La pondération permet d'affecter à chaque terme d'indexation une valeur qui mesure son importance dans le document où il apparaît. Nous aborderons cette notion en détail dans la section 7.1

Le processus d'indexation peut être illustré de manière simplifiée dans la figure suivante :

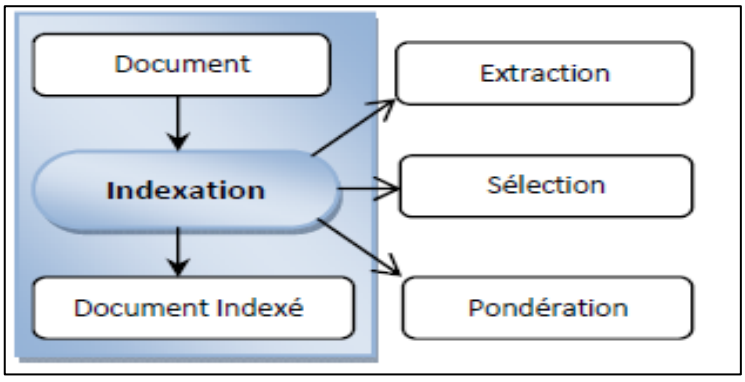



Figure 2 indexation d'un document *[14]*

### 6.2 Requêtage

Cette phase dépend de la représentation du document, les besoins d'information et les préférences de l'utilisateur (ex : la langue, la date, le format, etc.). Cette étape s'intéresse à l'expression des besoins de l'utilisateur, souvent à travers une liste de mots-clés représentant la requête **[15]**.Ainsi, la requête soumise par l'utilisateur subit les mêmes traitements que ceux réalisés précédemment sur les documents au cours de leur indexation. Toutefois, la requête peut être étendue ou reformuler pour renforcer les préférences des utilisateurs et le retour de pertinence **[16] [17]**. À la fin du processus de recherche, une liste de documents sera retournée.

### 6.3 Appariement

Une fois les documents indexés et la requête analysée, le système de RI procède à la mesure de pertinence de chaque document vis-à-vis d'une requête. Selon une fonction de correspondance relative au modèle de recherche, et à renvoyer ensuite à l'utilisateur une liste de résultats. Cette mise en correspondance génère un score de pertinence reflétant le degré de similarité entre la requête et le document. Ce score est calculé à partir d'une valeur appelée $\mathbf{RSV}(\mathbf{q}, \mathbf{d})$ (Retrieval Status Value), où $\boldsymbol{q}$ représente une requête et $\boldsymbol{d}$ un document. Le score final permet d'ordonner les documents retournés.

Il existe deux types d'appariement **[7]**:

- **Appariement exact :** Le résultat est une liste de documents respectant exactement la requête spécifiée avec des critères précis. Les documents retournés ne sont pas triés.

- **Appariement approché :** Le résultat est une liste de documents censés être pertinents pour la requête. Les documents retournés sont triés selon un ordre de mesure. Cet ordre reflète le degré de pertinence document/requête.

# 7 Fonctions de Recherche d'Information

La Recherche d'Informations utilise différentes méthodes et techniques de travail, indépendamment des modèles. Leur but est d'obtenir des résultats plus pertinents.

## 7.1 La pondération des mots

La pondération est une fonction fondamentale puisqu'elle traduit le degré d'importance des termes dans les documents. Parmi les nombreuses formules de pondération définies dans le domaine, la mesure TF-IDF est de loin la plus connue et utilisée.

Elle est basée sur la combinaison des deux facteurs : fréquence du terme (TF) et fréquence inverse de document (IDF). Elle est donnée par la multiplication des deux mesures TF et IDF

$$TF - IDF = TF * IDF$$

Les mesures TF et IDF sont définies comme suit :

### 7.1.1 TF (Term Frequency)

Cette mesure a été introduite pour tenir compte de la fréquence d'un terme dans un document. L'idée sous-jacente est que plus un terme est fréquent dans un document plus il est important dans sa description. Elle représente une "pondération locale" d'un terme dans un document.

En effet, le terme de recherche peut apparaître plus fréquemment dans un document long que dans un document court. Par conséquent, la fréquence doit être considérée par rapport à la taille d'un document. Pour ce faire, le nombre d'occurrences du terme recherché dans le document (la fréquence du terme) est divisé par le nombre total des termes dans le document.

$$\mathbf{TF} = \frac{le\ nombre\ d'occurence\ du\ terme\ t\ dans\ le\ document}{le\ nombre\ total\ des\ termes\ dans\ le\ document}$$

### 7.1.2 IDF(Inverse Document Frequency)

Ce facteur mesure la fréquence d'un terme dans toute la collection, c'est la "pondération globale". En effet, un terme fréquent dans la collection,

possède moins d'importance qu'un terme moins fréquent. Les mots que l'on ne trouve que dans très peu de documents, mais aussi très fréquemment, sont plus pertinents que ceux que l'on trouve dans presque tous les textes.

$$\mathbf{IDF} = \log\left(\frac{le\ nombre\ totat\ des\ document}{le\ nombre\ des\ document\ ou\ le\ terme\ t\ apparait}\right)$$

### 7.2 Mesures d'évaluation

En RI, la mise au point des modèles passe par une phase expérimentale qui suppose l'utilisation de métriques qui ont pour but de permettre la comparaison des modèles entre eux ou la mise au point de leurs paramètres. Cleverdon[7] a défini six critères qui peuvent être utilisés pour l'évaluation de la performance d'un SRI : la couverture de l'univers du discours de la collection, le temps de réponse, la présentation des résultats, l'effort requis de l'utilisateur pour retrouver parmi les documents retournés ceux qui répondent à son besoin, le taux de rappel et de précision du système.

Parmi ces critères, la précision et le rappel sont les plus populaires utilisés pour estimer l'efficacité du SRI exprimée par sa capacité à sélectionner tous les documents pertinents et à rejeter tous les documents non pertinents.

#### 7.2.1 Précision

Cette mesure calcule la capacité du système à rejeter tous les documents non pertinents pour une requête. Elle est donnée par le rapport entre les documents pertinents sélectionnés et l'ensemble des documents sélectionnés :

$$\mathbf{Précision} = \frac{l'ensemble\ des\ \text{les documents pertinents sélectionnés}}{l'ensemble\ des\ documents\ sélectionnés}$$

[7] Cyril Cleverdon : un britannique bibliothécaire et scientifique ordinateur qui est le mieux connu pour son travail sur l'évaluation des Systèmes de Recherche d'Information.

### 7.2.2 Rappel

Mesure la capacité du système à renvoyer tous les documents pertinents pour une requête. Il est donné par le rapport entre les documents pertinents sélectionnés et l'ensemble des documents pertinents pour la requête :

$$\textbf{Rappel} = \frac{l'ensemble\ des\ \text{les documents pertinents}\text{sélectionnés}}{l'ensemble\ des\ documents\ pertinents}$$

### 7.2.3 F-mesure

C'est une mesure qui combine la précision (P) et le rappel (R). Nommée F-mesure ou F-score, cette mesure fût introduite dans [18] et est définie par :

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

### 7.2.4 E-mesure

C'est la F-mesure paramétrique. Elle permet d'attribuer un ordre de préférence entre le rappel (R) et la précision (P). Définie par :

$$\textbf{E} - \textbf{mesure} = \frac{(1+\beta)PR}{\beta^2 P+R}$$

Si $\boldsymbol{\beta}$**= 1** : même poids précision et rappel

Si $\boldsymbol{\beta}$ **>1** : privilégie précision au rappel

Si $\boldsymbol{\beta}$**< 1** : plus d'importance au rappel

# 8 Les différents modèles de RI

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

Selon Baeza-Yates et Ribeiro-Neto [19], ce modèle est décrit par le quadruplet $(D, Q, F, R(q, d))$, où :

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

Il existe différents modèles de Recherche d'Information. Ils ne s'annulent pas nécessairement entre eux, mais peuvent au contraire être combinés. On peut les diviser en trois grandes catégories :

- **Les modèles ensemblistes**

Ces modèles trouvent leurs fondements théoriques dans la théorie des ensembles. On distingue le modèle booléen pure (Boolean Model), le modèle booléen étendu (Extended Boolean Model) et le modèle basé sur les ensembles flous (Fuzzy set model).

- **Les modèles vectoriels**

Basés sur l'algébrique, plus précisément le calcul vectoriel. Ils englobent le modèle vectoriel (Vector model), le modèle vectoriel généralisé (Generalized Vector Model), Latent Semantic Indexing (LSI) et le modèle connexionniste.

- **Les modèles probabilistes**

Se basent sur les probabilités. Ils comprennent le modèle probabiliste général, le modèle de réseau de documents ou d'inférence (Document Network) et le modèle de langue.

Nous présentons par la suite les principaux modèles issus de chacune de ces trois catégories :

### 8.1 Le modèle booléen

Sans contexte, ce modèle est considéré comme étant le premier modèle de RI [4]. Les moteurs de recherche les plus connus sur le Web sont basés sur le principe booléen. Dans ce modèle les documents sont représentés chacun par une conjonction de termes de la forme $d = t_1 \wedge t_2 \wedge \ldots \wedge t_n$ Ainsi, les requêtes sont représentées chacune par des expressions booléennes reliées par des opérateurs logiques (AND (∧), OR(∨), NOT (¬)). La fonction de pertinence $R(q, d)$ est définie pour indiquer la présence ou non des termes de la requête $q$ dans le document $d$. Parmi les inconvénients du modèle booléen : Pas d'ordre pour les documents sélectionnés, formulation de la requête difficile pas toujours évidente pour les utilisateurs non expérimentés.

### 8.2 Modèle vectoriel

Dans ce modèle, les documents et les requêtes sont représentés sous forme de vecteurs dans l'espace vectoriel engendré par tous les termes de la collection de documents (figure 3). Les termes sont cartographiés comme axes de coordonnées. Les deux vecteurs (vecteur de la requête et de document) sont ensuite comparés l'un à l'autre. Le vecteur le plus similaire à celui de la requête de recherche doit apparaître en premier dans le classement des résultats. L'inconvénient ici, c'est que la représentation vectorielle suppose l'indépendance entre termes.

La fonction de pertinence $RSV(q, d)$ est définie par : $\mathrm{RSV}(\mathrm{q}, \mathrm{d}) = \cos(\vec{q}; \vec{d})$.

Une collection de $n$ documents et $M$ termes distincts peut être représentée sous forme de matrice.

$$\begin{array}{c|cccc} & T_1 & T_2 & \cdots & T_M \\ D_1 & w_{11} & w_{21} & \cdots & w_{M1} \\ D_2 & w_{12} & w_{22} & \cdots & w_{M2} \\ \vdots & \vdots & \vdots & & \vdots \\ \vdots & \vdots & \vdots & & \vdots \\ D_n & w_{1n} & w_{2n} & \cdots & w_{Mn} \end{array}$$

Figure 3 Représentation de la collection sous forme de matrice

Tel que $w_{ij}$ concernent les poids des termes *T* dans les documents *D*.

La figure suivante illustre la représentation d'un modèle vectoriel

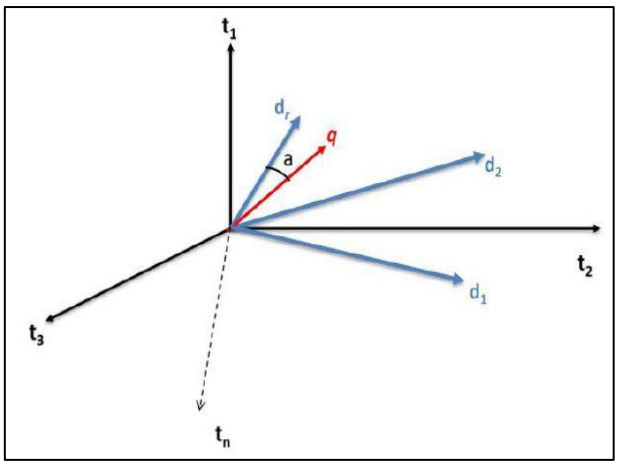



Figure 4 Représentation du modèle vectoriel

### 8.3 Modèle probabiliste

Dans ce modèle, la pertinence d'un document par rapport à une requête est donnée par un calcul de probabilité [20] [21]. Le principe de base de ce modèle est de trouver les documents qui ont une forte probabilité à être pertinents, et en même temps une faible probabilité à être non pertinents. La fonction de pertinence RSV (q, d) est donnée par la formule suivante :

$$\mathbf{RSV}(\mathbf{q},\mathbf{d}) = \sum_{i=1}^{n} log\frac{P(1-q)}{P(1-p)}$$

Où :

- p = P(terme $t_i$ présent | d pertinent).
- q = P(terme $t_i$ présent | d non pertinent).
- et n : le nombre de termes dans la requête.

La figure suivante résume quelques un des modèles de RI connus.

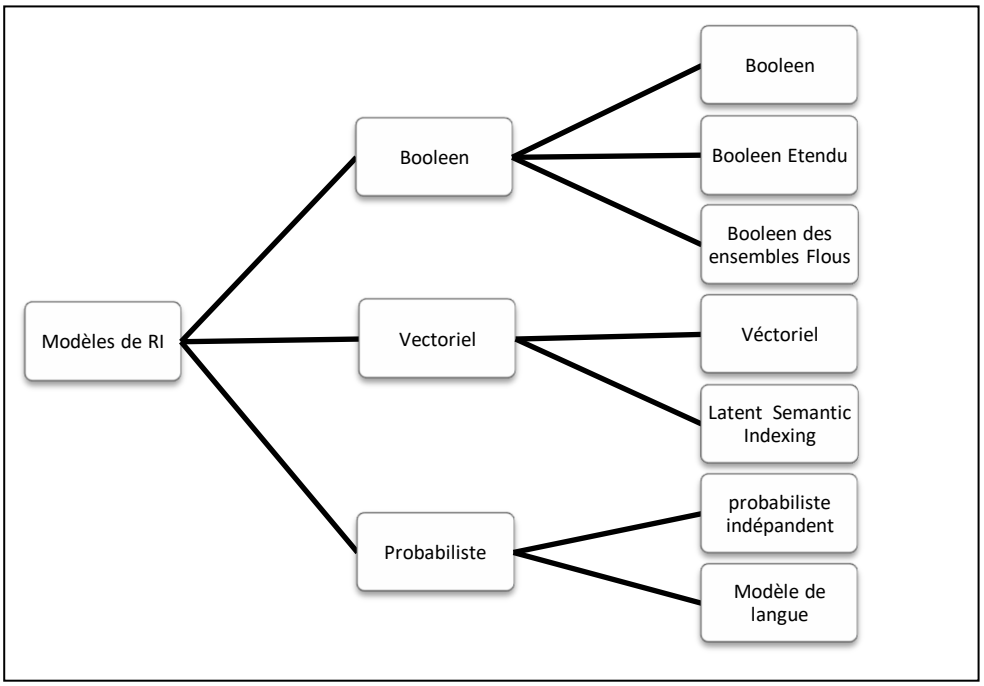



Figure 5 Modèles de la RI

Avec l'évolution de la RI, ces modèles ont évolué bien évidemment, et de nouveaux modèles sont nés de l'hybridation des modèles existants, comme le modèle qui est basés sur les Réseaux Bayésiens et les Réseaux Possibilistes [22], ainsi que d'autre modèles qui sont des extensions des modèles classiques de RI.

# 9 Conclusion

La Recherche d'Information Classique se base principalement sur le calcul de la pertinence du document selon des critères de sélection par le contenu et de la disponibilité de l'information ou alors elle peut également exploiter la structure des liens entre les documents afin de retourner une liste de résultats en réponse à une requête utilisateur.

Dans ce chapitre nous avons parlé du RI Classique, ces concepts de base, ces modèle et le processus de la RI, dans le prochain chapitre nous allons nous intéresser plus particulièrement aux opérations d'appariement document-requête et de classement des résultats.

# Chapitre II : L’appariement document-requête et le Classement en RI (Ranking)

# 1 Introduction

Étant donné une requête et une collection de documents qui correspondent à la requête, le problème est de classer, c'est-à-dire de trier, les documents selon certains critères afin que les «meilleurs» résultats apparaissent tôt dans la liste de résultats affichée à l'utilisateur. Parfois, nous avons besoin de classer les documents uniquement en fonction de leur pertinence par rapport à la requête. Dans d'autres cas, nous devons prendre en compte les relations de similarité et de diversité entre les documents dans le processus de classement (Ranking en Anglais).

Nous commencerons ce chapitre par la définition du Ranking, son principe et ces déférentes méthodes. Par la suite nous allons parler de différents types de mesures de similarités pour le calcul des appariements (mapping ou matching en Anglais) requête-document, en insistant sur les types que nous avons retenus dans notre travail.

# 2 Définition du Classement (Ranking)

Plusieurs définitions du classement ou Ranking ont été proposées dans la littérature. Parmi ces définitions [23]: le classement est la mise en ordre des rubriques selon des normes prédéfinies ; en outre, le Ranking en RI c'est la dernière phase du processus de Recherche. Où, le moteur de recherche retourne les documents classés selon un ordre de pertinence qui répond au besoin d'information des utilisateurs.

# 3 Principe du ranking

Un moteur de recherche utilise des algorithmes de classement pour produire la liste classée des documents selon leurs pertinences (dépendant de l'information qu'ils contiennent). Les caractéristiques de ces algorithmes modélisent des propriétés statistiques, ces dernières sont intéressées par le nombre d'occurrence du mot dans le document ou des structures linguistiques qui sont intéressées par le sens des mots, leur syntaxe ...etc.

Lorsque l'utilisateur émet une requête, la tâche d'un moteur de recherche est de décider si le document appartient à l'ensemble pertinent ou à l'ensemble non pertinent. Les documents pertinents sont ensuite classés selon leur degré de pertinence, leur importance, etc.

# 4 Méthodes de classement

Apres le calcul du score de similarité entre la requête et le document le processus de classement (Ranking) permet de classer les documents en fonction de leurs scores de similarité [24]. Les moteurs de recherche ont développé des méthodes de tri automatique des résultats. Cela leur permet ainsi de se distinguer les uns des autres. Dans la pratique, aucune méthode de tri n'est parfaite mais cette variété offre à l'utilisateur la possibilité de traquer l'information de différentes manières. Elle augmente donc ses chances d'améliorer ses recherches.

Le but du classement est d'afficher dans les dix à vingt premières réponses les documents répondant le mieux à la question. On peut considérer trois grandes méthodes de tri [24]:

## 4.1 Tri par pertinence

C'est une méthode d'affichage des résultats de la requête selon un ordre qui est déterminé par le calcul d'un score pour chaque réponse. Cette méthode repose sur des travaux de recherche déjà anciens de Robertson et Sparckjones [25] , implémentés dans le logiciel d'indexation WAIS à la fin des années 80.

L'estimation de pertinence est basée sur les critères suivants appliqués aux termes de la requête :

- **La fréquence d'occurrence du terme dans la base de données** (poids d'un terme en fonction du nombre d'occurrences) : élimination des mots-vides, pondération des mots rares ou peu fréquents.

- **la densité du terme** : calculée en fonction du rapport entre l'occurrence du terme dans le document et la taille du document : si deux documents ont la même occurrence pour le même terme, le document plus petit sera favorisé en pondération (Exemple : si le mot "bibliothèque" apparaît 10 fois dans deux documents, l'un de trois pages, l'autre de 50 pages, le document de trois pages sera jugé plus pertinent)

- **la position du terme dans le texte :** le poids d'un terme dans un document est déterminé par sa place dans le document : il est maximum pour le titre et le début du texte; à l'intérieur, il est plus important si le mot est en majuscule .

- **la similarité des termes du document avec les termes de la requête :** correspondance exacte ou partielle des mots et la relation de proximité, elle est basée sur la proximité des termes de la requête entre eux dans le document. Le degré de proximité des termes dans le document induit un poids plus élevé. Dans notre travail, nous nous sommes particulièrement intéressées à cette méthode de tri. A cet effet, plus de détails sur les mesures de similarité seront élaboré par la suite.

### 4.2 Tri par popularité

C'est une méthode fondée sur la prise en compte, non plus du contenu, mais de la spécificité du Web : les hyperliens.

Le tri par popularité recouvre deux méthodes :

- **Méthode fondée sur la Co-citation :** Algorithme d'évaluation de pertinence fondé sur la nature même du web, c.-à-d. son hyper textualité : les algorithmes vont donc explorer les réseaux de documents et de liens qui relient les documents. Les pages affichées en premier sont les pages référencées de nombreuses fois.

- **Méthode fondée sur la mesure d'audience :** propose de trier les pages en fonction du nombre de visites qu'elles reçoivent (indice de clic), c.-à-d. l'analyse du comportement de l'internaute lors de la Recherche d'Information et de l'utilisation du moteur de recherche; qui vise à trouver les pages les plus populaires.

### 4.3 Tri par calcul dynamique de catégories

Méthode de clustering ou agrégation ; développée en bibliométrie dans les années 80 et appliquée à des corpus documentaires pour la veille technologique. Actuellement, prise en compte par les outils de Text Mining; utilisation de catégories prédéfinies et de catégories repérées automatiquement.

## 5 Appariement requête-document

Comme mentionné dans la section précédente, il y'a 3 types de classement des documents pertinents, parmi lesquels la méthode de tri par pertinence, elle-même basée sur différents critères. Dans notre travail, nous allons nous intéresser à un de ces critères qui est la similarité des termes du document avec les termes de la requête. Une explication plus détaillée de ces mesures est donnée ci-dessous.

### 5.1 Similarité Syntaxique

L'analyse syntaxique est l'étude de la structure de la phrase. Les méthodes de cette mesure sont basées sur la comparaison de mots, de chaînes de caractère ou de texte et sur les lettres qu'ils ont en commun.

Il existe plusieurs mesures de similarité syntaxiques citons : la distance de *Levenshtein* (ou distance d'édition), le coefficient de *Dice*, l'indice de *Jaccard*, la distance *euclidienne*…etc. Dans notre cas, nous utilisons la distance *Jaro* pour la similarité syntaxique et le cosinus .

- **La distance Jaro** est particulièrement adaptée à la comparaison des chaines courtes et sera donc parfaite pour le mapping des descriptions de documents et les mots-clés des requêtes. Le résultat est normalisé de façon à avoir une mesure entre 0 et 1 (zéro représente l'absence de similarité et 1, l'égalité des chaines comparées). La distance entre deux chaine S1 et S2 est définit comme suit [13] :

$$synSim = \frac{1}{3}\left(\frac{m}{|S_1|} + \frac{m}{|S_2|} + \frac{m-t}{m}\right)$$

Avec :

- **m** est le nombre de caractères correspondants.
- **t** est le nombre de transpositions (obtenu en comparants le i-ème caractère correspondant de S1 avec le i-ème caractère correspondant de S2. Le nombre de fois où ces caractères sont différents, divisé par deux).
- $|S_i|$ est la longueur de la chaine de caractère $S_i$.

Deux caractères identiques de S1 et S2 sont considérés comme correspondants si leur éloignement (i.e. la différence entre leurs positions dans leurs chaines respectives) ne dépasse pas [4]:

$$\left(\frac{max(|S_1|, |S_2|)}{2}\right) - 1$$

- **Le cosinus** : La similarité cosinus est fréquemment utilisée en tant que mesure de ressemblance entre deux documents d1 et d2, quelle que soit leur taille. Mathématiquement, on mesure le cosinus de l'angle entre deux vecteurs projetés dans un espace multidimensionnel. Plus l'angle est petit, plus la similitude cosinus est élevée. La similarité obtenue simCosinus(d1, d2) ∈ [0, 1]. Elle est calculée avec la formule suivante :

$$simCosinus = \frac{\sum_{i=1}^{n} x_i \times y_i}{\sqrt{\sum_{i=1}^{n} x_i^2 \times \sum_{i=1}^{n} y_i^2}}$$

Où : -$x_i$ : poids du terme dans le document

- $y_i$ : poids du terme dans la requête

### 5.2 Similarité lexicale

C'est la mesure du degré de ressemblance entre des séries de mots appartenant à deux langages donnés.

Cette dernière utilise des méthodes nécessitant l'utilisation de ressources externes. Plusieurs types de ressources peuvent être utilisés, mais nous avons choisi WordNet.

WordNet est une grande base de données lexicale pour la langue anglaise où les noms, les verbes, les adjectifs et les adverbes sont regroupés dans des ensembles de synonymes cognitifs (appelés synsets). Les synsets sont liés par moyens de relations conceptuelles sémantiques et lexicales.

La formule de similarité lexicale entre S1 et S2 se calcule ainsi [13] :

$$lexSim(S_1, S_2) = \frac{\beta}{min(|Syn(S_1)|, |Syn(S_2)|)}$$

Où : min ($|syn(S1)|$, ($|syn(S2)|$) le minimum des cardinalités de deux ensembles $|syn(C1)|$ et $|syn(C1)|$ et $\beta=(|syn(S1) \cap syn(S2)|$.

Cette mesure renvoi 1 si au moins S1 et S2 ont 1 synset commun. 0 est retournée dans le cas ou S1 et S2 ne sont pas synonymes et n'ont pas de relation lexicale (antonymes, hyponymes…).

### 5.3 Similarité Terminologique

Cette mesure est employée pour calculer la valeur de similarité des entités textuelles, telles que des noms, des méta-données[8] sur les noms, des commentaires,…etc. Elle est basée sur une simple comparaison de chaines de caractères est n'exploite pas la signification des termes.

Cette dernière utilise les résultats du calcul de la similarité lexicale et syntaxique, tels que la formule de combinaison est [13] :

$$terSim(S_1, S_2) = \frac{(lexSim(S_1, S_2) \times lexCoeff) + (synSim(S_1, S_2) \times synCoeff)}{(lexCoeff + synCoeff)}$$

Ou Coeff est un coefficient numérique calculé comme suit : Coeff=$exp^{sim}$

### 5.4 Similarité Structurelle

Les méthodes de similarité structurelle déduisent la similarité de deux mots, en utilisant des informations structurelles, les méthodes calculent la similarité entre deux concepts (mots, chaînes de caractère.) en utilisant soit des informations sur leur structure interne, ou bien externe utilisant la structure hiérarchique d'une ressource lexicale. Et ce, en comptant le nombre d'arcs dans la hiérarchie pour déterminer la similarité entre deux entités. La similarité de **Wu et Palmer** est une mesure qui utilise également la notion du Plus Court Chemin (PCC) entre les concepts mais dépend aussi de leur position dans le réseau. En effet, la mesure de Wu et Palmer a tendance à exprimer un éloignement sémantique pour des concepts proches de la racine. Ainsi pour deux concepts C1 et C2, Wu et Palmer calcule leur similarité de la manière suivante [13]:

$$Sim(C_1, C_2) = \frac{2 \times PCC(LCS(C_1, C_2), RACINE)}{PCC(LCS(C_1, C_2), C_1) + PCC(LCS(C_1, C_2), C_2) + 2 \times PCC(LCS(C_1, C_2), RACINE)}$$

[8] Les métadonnées sont les données qui décrivent d'autres données ; Par exemple les métadonnées les plus courantes sont la date de sauvegarde, la taille et l'auteur du fichier…etc.

Où PCC(c1,c2) est le plus court chemin de deux concepts est le nombre minimal d'arêtes pour aller d'un concept à un autre.

Le LCS (Least Common Subsumer) de deux concepts est le concept hyperonyme à ces deux concepts.

### 5.5 Similarité Sémantique

Une mesure de similarité sémantique est un concept selon lequel un ensemble de documents ou de termes se voient attribuer une métrique basée sur la ressemblance de leur signification/ contenu sémantique [26]. Deux concepts sont considérés comme sémantiquement similaires s'il y a une synonymie, hyponymie[9], antonymie, ou toponymie[10] entre eux (Exemples : Médecin-Chirurgien, Sombre-Clair). Deux sens de mots sont considérés comme sémantiquement liés s'il existe au moins une relation lexico-sémantique entre eux - classique ou non classique (Exemples : Chirurgien-Scalpel, Arbre-Ombre) [27].

Sa formule est la suivante :

$$semSim(S_1, S_2) = \frac{(terSim(S_1, S_2) \times terCoeff) + (strSim(S_1, S_2) \times strCoeff)}{(terCoeff + strCoeff)}$$

Après ces étapes, nous obtenons une matrice de corrélation, où les cellules contiennent des mesures de similarité. A partir de cette matrice, un vecteur de similarité simVecrtor est généré, plus de détails sur cette notion seront élaboré dans le chapitre conception.

[9] Relation sémantique hiérarchique décrit une relation is-a typique, indiquant qu'un concept subsume un autre.

[10] Relation sémantique entre deux verbes, l'un décrivant de manière plus précise l'action de l'autre. Le premier verbe est dit toponyme du second.

La figure suivante montre un exemple de calcule des similarités précédente :

```
Term 1 love

Term 2 hate

***Les similaritées***
Syntxique : 0.5
lexical 0.0
Terminologique 0.3112296656009273
Structurelle 0.8571428571428571
Sémantique 0.6568947478270274
```

Figure 6 Exemple de similarité

# 6 Travaux antérieurs lié au projet

1. **Relevance Ranking Based on Query-Aware Context Analysis** [28]

   - **Problématique :** Dans les travaux récents, les représentations de mots distribués résolvent le problème de la non-correspondance des mots en permettant la correspondance sémantique. Cependant, la plupart des modèles de classement des documents existants sont basés sur la correspondance sémantique entre la requête et les termes du document sans une compréhension explicite de la relation de la correspondance et la pertinence.
   - **Proposition :** Ils ont proposé d'utiliser des contextes locaux des termes de la requête dans les documents pour la correspondance sémantique. La mise en correspondance faite avec des petits contextes de requêtes qui sont un résultat enregistré par un processus de jugement. Ce dernier a été fait par des observateurs humains.
   - **Dataset :** Ils ont utilisé trois collections standards de TREC dans leurs expériences qui sont : Robust, Gov2, et WT10g.

     Ils ont utilisé le titre du sujet comme une requête, la liste standard INQUERY des mots vides pour enlever les mots vides et aucun stemming n'a été effectué.

- **Evaluation :** Ils ont comparés leur méthode proposée (LCD-Logistic) avec les lignes de base (baselines) SDM (dans la base de donnée Robust) . Ils ont rapporté également GMAP pour évaluer leur méthode face à des requêtes difficiles. Selon les résultats obtenus, LCD-Logistic améliore considérablement le rappel dans les collections Robust et Gov2.
- **Résultat :** Apres la comparaison entre les variantes LCD et LCA du modèle proposer ,.. montre que la variante LCD dépasse LCA dans tous les cas. Ça veut dire que l'utilisation du contexte local le plus pertinent du document pour faire son score ,suivi le principe de pertinence disjonctive donne la meilleure performance .

2. **Context-aware information retrieval systems: contribution to a semantically enriched, folksonomy-based text-search** [13]
   - **Problématique :** Parmi les étapes du processus du RI, on a l'indexation et plus particulièrement le Stemming. Ce dernier consiste en la transformation d'un mot en Stem (radical), mais cela peut produire des unités qui n'ont pas de sens. Ceci peut influencer les résultats retournés aux utilisateurs (nous nous sommes contentées de parler d'un seul problème posé dans ce travail. Celui qui rentre dans le cadre du travail que nous avons entamé).
   - **Proposition:** dans ce travail, une méthode a été proposée pour atténuer les problèmes de l'approche traditionnelle de stemming .

     Cette méthode est basée sur une combinaison de stripping d'affixes (basé sur l'algorithme Porter Stemming), de techniques contextuelles (basées sur l'algorithme Context-Aware Stemming «CAS») et de techniques de lemmatisation basées sur un corpus pour la langue anglaise (basée sur WordNet).
   - **Dataset**: les auteurs ont utilisé l'ensemble de données WT2G[11] (de la collection TREC) avec la plate-forme Terrier pour prouver

[11] URL : http://ir.dcs.gla.ac.uk/test_collections/access_to_data.html

l'efficacité de leur algorithme de Stemming, de récupération et de classement.

- **Evaluation :** Pour évaluer leur algorithme d'indexation et comparer les résultats obtenus avec l'algorithme de Stemming Porter et l'algorithme CAS, ils ont utilisé la plateforme Terrier[12] pour évaluer les fichiers finaux.
- **Résultat :** Des résultats très encourageants ont été obtenus, non seulement grâce à l'algorithme Stemming, mais aussi grâce à l'amélioration significative obtenue par la nouvelle méthode de mapping des requêtes et documents. Ainsi, afin d'approfondir l'évaluation de l'algorithme de Stemming, nous avons analysé la compression du corpus et avons constaté que sa taille était réduite à un tiers (1/3) grâce à notre algorithme de Stemming et sa phase de normalisation.

Le tableau suivant (Tableau 1), est un tableau qui résume les avantages de chacun des travaux mentionnés ci-dessus.

| **Relevance Ranking Based on Query-Aware Context Analysis** | **Context-aware information retrieval systems: contribution to a semantically enriched, folksonomy-based text-search** |
|---|---|
| • Ce modèle est basé sur les jugements humains pour trouver des scores de similarité de haute qualité entre la requête et le document.<br>• La méthode proposée est conçue pour être capable de capturer des heuristiques IR importantes, par exemple, la proximité des termes de requête | • Ce modèle fonctionne bien avec des documents volumineux.<br>• Tous les stems sont des mots valides car une base de données lexicale qui fournit des formes précises pour les mots est utilisée dans le processus de radicalisation.<br>• Il a été prouvé qu'il donnait de meilleurs résultats que le CAS et l'algorithme Porter pour l'indexation. |

[12] http://terrier.org/

| | |
|---|---|
| dans les documents, la correspondance sémantique entre les termes de requête et de document, et l'importance des termes de requête.<br>• Ce modèle peut être intégré dans n'importe quel modèle de récupération et améliorer considérablement leurs performances. | • La technique d'appariement proposée a prouvé de bons résultats aussi bien dans le domaine de la fusion des ontologies que le matching document-requête. |

Tableau 1 Avantages des Travaux

# 7 Conclusion

Dans ce chapitre, nous avons parlé de la dernière phase du processus de la RI, qui est le classement des documents, nous avons parlé de son principe et ses méthodes.

Nous avons aussi présenté les différentes mesures de similarité que nous utiliserons pour la réalisation de notre travail dont nous allons parler dans les deux chapitres suivants.

# Chapitre III : Conception et modélisation de la solution

# 1 Introduction

Dans ce chapitre, nous allons parler de notre proposition et sa conception. Nous allons entamer le chapitre en présentant notre architecture globale. Après, nous allons passer en revue chaque composants de la solution en expliquant leur logique de fonctionnement, commençant par le Dataset utilisé et sa structure, passant au prétraitement effectué et quelques notions sur les Réseau de Neurones, en insistant sur le MLP comme type de réseaux de neurones retenu pour notre solution, suivi du fonctionnement du modèle neuronal.

# 2 Architecture globale

Dans cette section, nous présentons l'architecture globale de notre système. Elle inclut tous les composants, les traitements, et les interactions (utilisateur-système). Cette architecture est illustrée par la figure ci-dessous (figure 8).

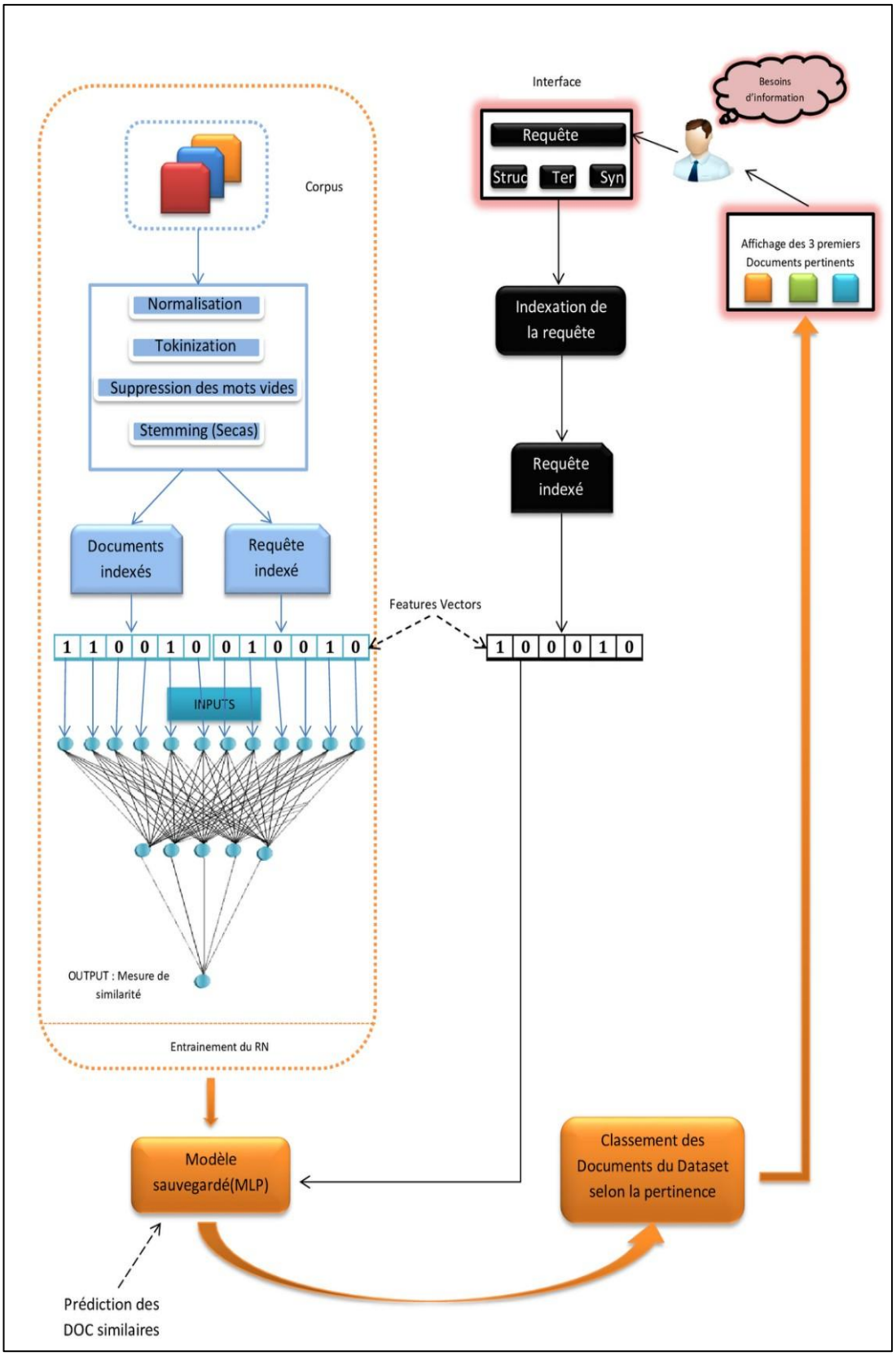



Figure 7 Architecture global du système

Comme illustré ci-dessus, le système est composé de deux parties importantes. D'un côté réseau de neurones et son entrainement pour aboutir à un modèle bien entrainé pour les tests et de l'autre côté, nous avons l'utilisateur qui émet sa requête, et choisit la mesure de similarité qu'il lui convient. Après un traitement, et des prédictions faites par le Réseau de neurones. Les trois premiers documents qui répondent le plus au besoin de l'utilisateur seront retournés.

# 3 Description de l'architecture du Système

Dans les sections qui suivent, nous décrivons en détail tous les composants et les traitements du système.

## 3.1 Dataset (Corpus)

Les entrées du Réseau de Neurones sont des feature vectors du document indexé ainsi que la requête. Fautes de temps, nous n'avons pas trouvé un dataset prêt qui répond à nos besoins et avec le format qui nous convient. Du coup, nous avons créé notre propre dataset pour le training du Réseau de Neurone.

Le dataset est de format CSV[13], il repose sur trois composants importants : La requête, le document et la mesure de similarité (requête-document), il contient 150 (requête, document et similarité (req/doc)), avec une taille de 79.5 Ko, la majorité des couples (req+doc) ont une relation avec le domaine informatique, plus d'autre couples pris de la collection NLP (aussi connu sous le nom de VASWANI)[14].

Le dataset est composé de 8 champs :

-Query

-Document

[13] CSV (Comma-Separated Values) : un format texte ouvert représentant des données tabulaires sous forme de valeurs séparées par des virgules.

[14] http://ir.dcs.gla.ac.uk/resources/test_collections/npl/

-6 champs pour chaque similarité utilisée : Similarité Cosinus, Syntaxique, Lexicale, Terminologique, Structurelle et Sémantique.

Un extrait d'une cellule du dataset est présenté dans la figure suivante :

| | A |
|---|---|
| 1 | Query,Document,Similarity,Syntaxique,Lexical,Terminologique,Structurelle,Semantique |
| 2 | Neural network ,A neural network is a series of algorithms that endeavors to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates In this sense neural networks refer to systems of neurons either organic or artificial in nature Neural networks can adapt to changing input so the network generates the best possible result without needing to redesign the output criteria The concept of neural networks which has its roots in artificial intelligence is swiftly gaining popularity in the development of trading systems Artificial neural networks ANNs usually simply called neural networks NNs are computing systems vaguely inspired by the biological neural networks that constitute animal brains An ANN is based on a collection of connected units or nodes called artificial neurons which loosely model the neurons in a biological brain Each connection like the synapses in a biological brain can transmit a signal to other neurons An artificial neuron that receives a signal then processes it and can signal neurons connected to it The signal at a connection is a real number and the output of each neuron is computed by some non linear function of the sum of its inputs The connections are called edges Neurons and edges typically have a weight that adjusts as learning proceeds The weight increases or decreases the strength of the signal at a connection Neurons may have a threshold such that a signal is sent only if the aggregate signal crosses that threshold Typically neurons are aggregated into layers Different layers may perform different transformations on their inputs Signals travel from the first layer the input layer to the last layer the output layer possibly after traversing the layers multiple times,0.13,1.0,1.0,1.0,0.5,0.8655292893150024 |

Figure 8 Un extrait du Dataset

## 3.2 Prétraitement

- **Normalisation** : C'est un processus (ensemble des techniques) qui rend les séquences des mots plus uniformes (exemple figure 10)

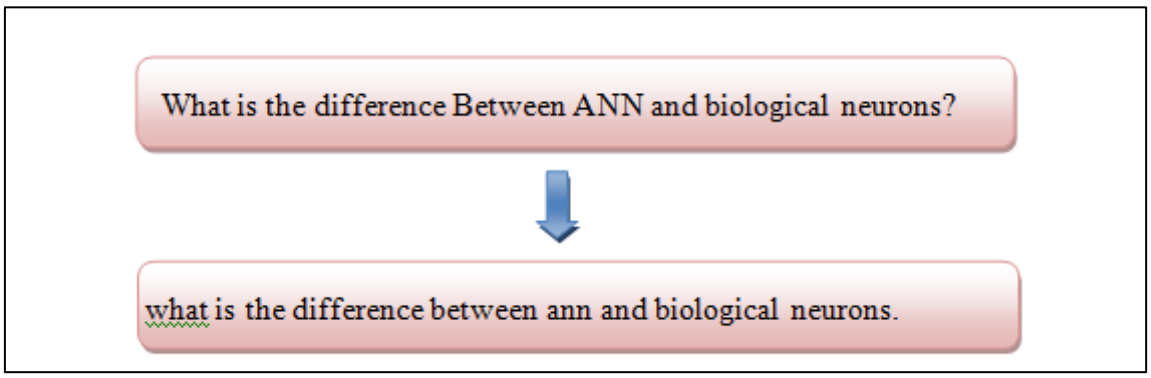



Figure 9 Normalisation.

- **Tokenization** : ce processus consiste à segmenter le texte de document en mot, en enlevant les espaces blanc et les caractères de ponctuation (exemple figure 11).

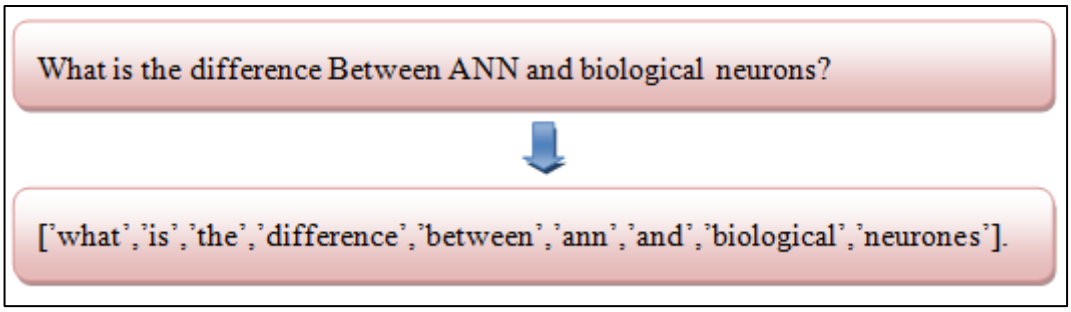



Figure 10 Tokenization

- **Suppression des mots vide** : dans cette phase nous allons Supprimer tous les mots vides (pronoms personnels, prépositions…etc.) (Exemple figure 12).

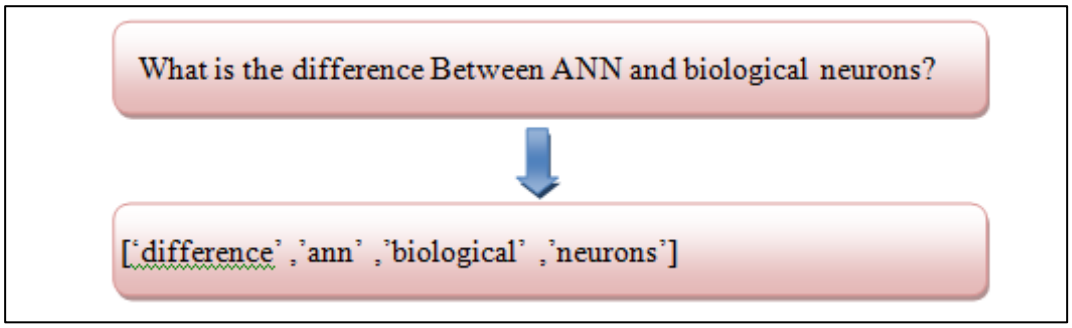



Figure 11 Suppression des mots vide

- **Stemming** : dans cette étape nous avons utilisé l'algorithme SECAS (Semantically Enriched Context-Awere Stemming) pour obtenir de bons stem. Parce que c'est le meilleur algorithme selon les résultats obtenu dans [13] (exemple figure 13).

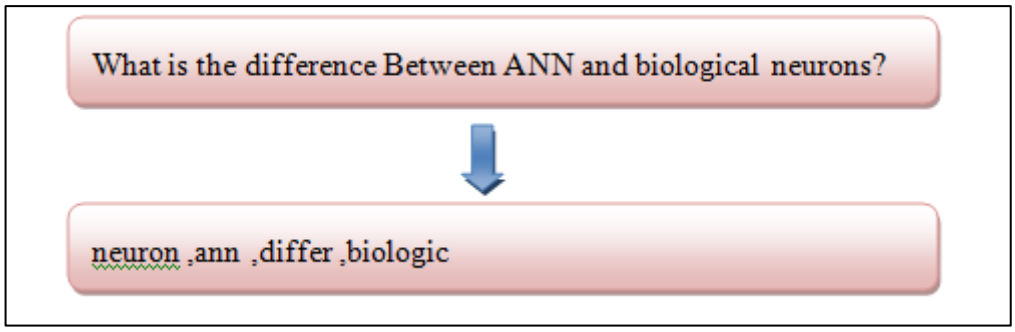



Figure 12 Stemming avec SECAS

### 3.3 Indexation

L'indexation est un processus consistant à reformuler le contenu d'un document sous une forme plus adaptée à son exploitation dans une application donnée, Chaque mot d'un document ou d'une requête obtenu après le prétraitement passe par une indexation avec l'algorithme : SECAS [13] Pour obtenir sa forme de base (racine), en se basant sur un modèle de RI vectoriel.

L'algorithme SECAS a été proposé pour atténuer les problèmes de l'approche traditionnelle de stemming qui effectue une transformation aveugle de tous les termes de requête et de document sans tenir compte du contexte du mot dérivé pour une recherche efficace en ce qui concerne la sensibilité au contexte [13].

### 3.4 Features vector

Dans l'apprentissage automatique, les vecteurs de caractéristiques (ou feature vectors en Anglais) sont utilisés pour représenter les caractéristiques numériques ou symboliques d'un objet de manière mathématique et facilement analysable. Les algorithmes d'apprentissage automatique nécessitent généralement une représentation numérique des objets pour que les algorithmes puissent effectuer le traitement et l'analyse statistique, dans notre cas les feature vectors des documents et requêtes vont être les entrées du Réseau de Neurone.

La figure suivante montre un exemple d'extraction d'un feature vector pour une requête donnée « neural network ». Tel que le vecteur de la requête (Query vector) et du document (Document vector) sont remplit par les occurrences des termes de la requête. Dans les deux cas, nous fixons la taille du feature vector à 8 (supposons que c'est la taille de la plus grande requête qu'un utilisateur peut émettre). Ce dernier sera remplit en se basant sur différents critères, et le reste du vecteur sera remplit par défaut avec des zéros.

```
Vocabulary :  {'neural': 1, 'network': 0}
Query vector :  2 [1 1]
Document vector :  2 [7, 2]
Feature Vector  [9, 0.0, 2, 7, 2, 0, 0, 0]
```

Figure 13 Feature Vector.

### 3.5 Réseau de Neurone

Le concept de réseaux de neurones artificiels (Artificial Neural Networks ANN) a été inspiré par les neurones biologiques. Dans un réseau de neurones biologiques, plusieurs neurones travaillent ensemble, reçoivent des signaux d'entrée, traitent des informations et déclenchent un signal de sortie. Les neurones biologiques sont groupés dans différentes couches et transmettent des signaux de proche en proche. Ces signaux contiennent des informations pouvant nous aider à déterminer des modèles, à identifier des images, à calculer des nombres et à prendre des décisions tout au long de notre vie [29] .

Le réseau de neurones en intelligence artificielle est basé sur le même modèle que le réseau de neurones biologique (figure 15). Bien que le concept sous-jacent soit le même que celui des réseaux biologiques, le réseau de neurones artificiel est un groupe d'algorithmes mathématiques produisant une donnée de sortie (output) à partir des données d'entrée (input).

Dans un réseau de neurones artificiels, plusieurs algorithmes travaillent ensemble pour effectuer des calculs sur les données d'entrée afin de produire une donnée de sortie. Ces données de sortie peuvent également aider le réseau de neurones à apprendre et à améliorer leur précision.

Les réseaux de neurones sont entrainés avec une multitude de données d'entrées couplées à leurs données de sortie respectives. Ils calculent ensuite la donnée de sortie en la comparant à la donnée de sortie réelle connue et se mettent à jour en permanence pour améliorer les résultats (si nécessaire). Au cours du temps, la donnée de sortie est utilisée pour améliorer la précision du modèle de notre réseau de neurones [30] .

Les réseaux de neurones peuvent aider les machines à identifier des modèles, des images …etc.

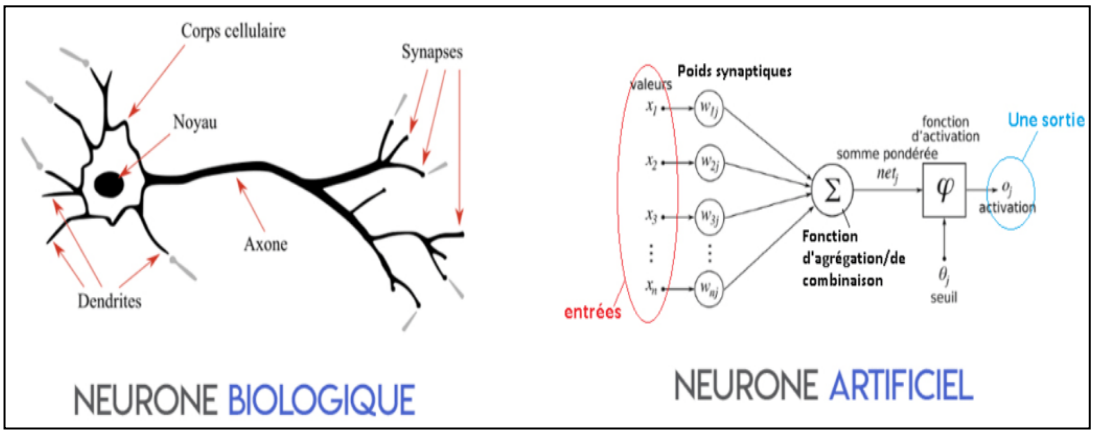



Figure 14 Différence entre Neurone Biologique et Neurone Artificiel

Il existe différents types de Réseau de Neurone [29] citons Les **réseaux de neurones récurrents (RNN)**, Les **réseaux de neurones convolutifs (CNN)** et les réseaux de type **Multi-layer Perceptron (MLP).**

Dans ce qui suit, nous présentons un tableau comparatif entre les trois types cité précédemment.

| RN | Architecture | Utilisation |
|---|---|---|
| ***RNN*** | Ce sont des réseaux de neurones spécialisés qui utilisent le contexte des entrées lors du calcul de la sortie. La sortie dépend des entrées et des sorties calculées précédemment.<br>ils utilisent des boucles de rétroaction pour traiter une séquence de données qui façonne le résultat final, lui-même pouvant être une séquence de données. Ces boucles de rétroaction permettent aux informations de persister, effet souvent assimilé à la mémoire. [31] | Les RNN conviennent aux applications où les informations historiques sont importantes. Ces réseaux nous aident à prévoir les séries chronologiques dans les applications commerciales et à prévoir les mots dans les applications de type chatbot, ont tendance à intervenir dans des modèles linguistiques visant à identifier la prochaine lettre d'un mot ou le prochain mot d'une phrase d'après les données qui les précèdent. Ils peuvent fonctionner avec différentes |

| | | |
|---|---|---|
| | | longueurs d'entrée et de sortie et nécessitent une grande quantité de données. |
| ***CNN*** | Ces réseaux reposent sur des filtres de convolution (matrices numériques). Les filtres sont appliqués aux entrées avant que celles-ci ne soient transmises aux neurones. Les CNN ont une méthodologie similaire à celle des méthodes traditionnelles d'apprentissage supervisé : ils reçoivent des images en entrée, détectent les features de chacune d'entre elles, puis entraînent un classifieur dessus. [32] | Ces réseaux de neurones sont utiles pour le traitement et la prévision d'images, reconnaissance faciale, numérisation de textes, traitement naturel du langage. |
| ***MLP*** | Les MLP sont des algorithmes de machine learning supervisé. Ils prennent la description d'un objet en entrée, et fournissent une prédiction en sortie. L'entrée est représentée par un vecteur numérique, qui décrit les caractéristiques (features) de l'objet. Ce vecteur traverse une succession de couches de neurones, où chaque neurone est une unité de calcul élémentaire. La prédiction est fournie en sortie sous la forme d'un vecteur numérique. | Les MLP sont utilisable pour les problème de prédiction de Regression où une quantité à valeur réel est prédite à partir d'un ensemble d'entrées , ils sont aussi utilisable pour la classification des documents. |

Tableau 2 Comapraison entre les RN

Dans notre architecture nous avons utilisé un réseau de neurone de type MLP , vu que nous somme entraine de résoudre un problème de régression ,et les valeurs prédite sont des valeurs réelles qui sont la mesure de similarité entre le document et la requête ,la figure 6 illustre une simple architecture d'un MLP.

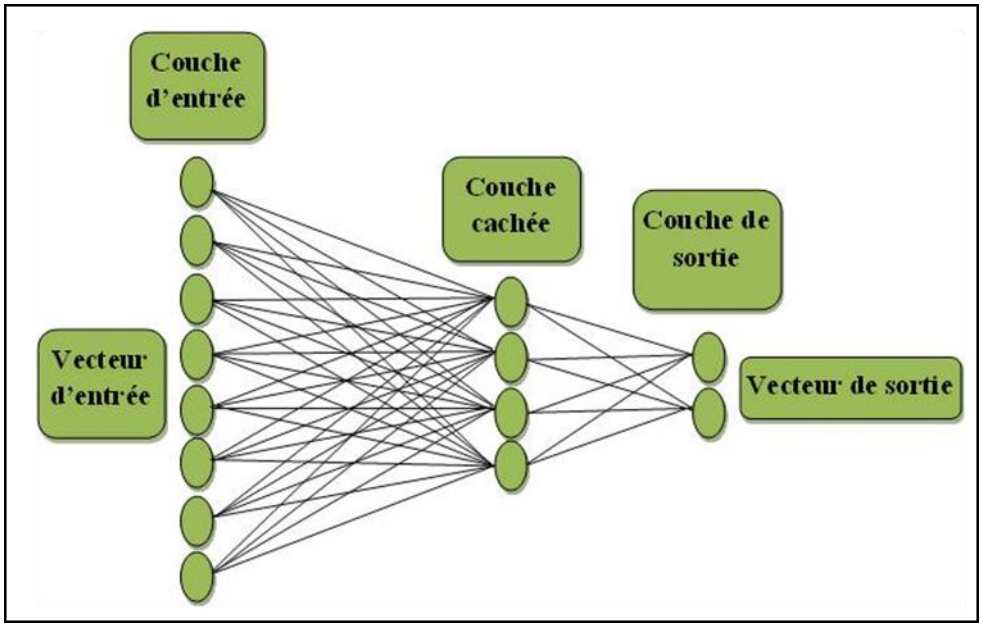



Figure 15 Un Réseau de Neurone MLP

### 3.6 Fonctionnement de notre modèle neuronal

Dans notre réseau, nous avons utilisé un MLP pour faire les prédictions car Il permet de réaliser des fonctions complexes de classification. Ce type de réseaux de neurones est dit supervisé, parce qu'il nécessite la description d'un résultat pour être en mesure d'apprendre. Le Réseau est composé de 300 neurones en entrées (Inputs). Tels que les inputs sont les features vectors des documents et requêtes. Pour l'architecture des couches cachés (hidden Layers) c'est très compliqué de savoir qu'elle configuration donnera la meilleure capacité d'apprentissage au réseau, quant à notre réseau il est composé de 5 couches cachées où le nombre de neurones diminuera de 50 nœuds dans chaque couche jusqu'à l'arrivée à la couche de sortie (Output layer).

Allons plus dans les détails sur les plus petites pièces du réseau de Neurones. Chaque neurone reçoit des informations numériques en provenance du neurone voisin. A chacune de ces valeurs est associé un poids représentatif de la force de connexion. Lorsqu'une donnée d'entrée entre dans un neurone, le poids sur le neurone est multiplié par sa valeur d'entrée. Ainsi, nous calculons la somme des poids multipliés par les valeurs d'entrée à laquelle on ajoute le biais. Enfin, une fonction d'activation est appliquée à cette somme pondérée. Notant que cette fonction est la fonction mathématique qui permet de traiter l'information qui arrive à un neurone artificiel en machine learning ; comme le fait le cerveau avec les signaux électriques qu'il reçoit.

On distingue différents type de fonctions d'activation tels que : Sigmoïde, Fonctions de tangente hyperbolique (tanh), Unité linéaire rectifiée (ReLu) et Softmax , en ce qui nous concerne, nous avons utilisé la fonction ReLu Cette fonction converge plus rapidement, optimise et produit la valeur souhaitée plus rapidement. C'est de loin la fonction d'activation la plus populaire utilisée dans les couches cachées, elle est de la formule suivante :

$$f(x) = max(0, x)$$

Elle donne une sortie *x ;* si *x* est positif et 0 sinon.

La valeur de sortie d'un neurone peut ensuite être renvoyée aux neurones de la couche suivante ce qui peut aider les réseaux de neurones à modifier le poids de leurs neurones.

Chaque neurone effectuera deux calculs : Addition linéaire des entrées, et fonction d'activation, comme illustré dans la figure suivante :

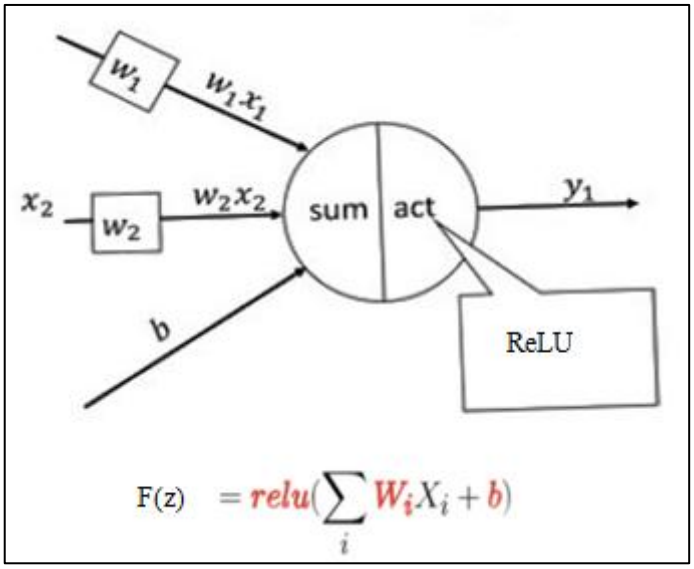



Figure 16 Fonction d'activation ReLU.

Où :

- $Y_1$ =f(z) ; (f(z) fonction d'activation).
- $z=w_1 x_1 + w_2 x_2 +b$ ; (« x » les inputs, « w » les poids et « b » le bias).

Et c'est après beaucoup d'exemples, et de correction des poids que notre Réseau de Neurone est devenu capable d'interpréter les similarités sans se tromper. Après plusieurs itérations, nous aboutissons à un modèle bien entrainé capable de prédire la similarité. Nous l'avons sauvegardé pour effectuer nos tests. Notons que l'entrainement du réseau a été effectué sur 70% des données du dataset et le reste (30%) pour les tests.

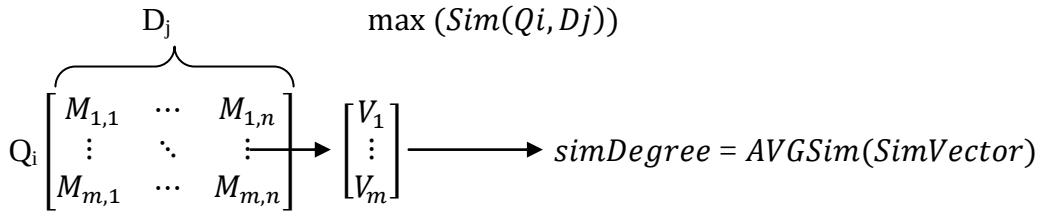
### 3.7 Matrice de corrélation

Après le calcul des similarités, nous obtenons une matrice de corrélation, où les cellules contiennent des mesures de similarité.

Soit *M* la matrice de corrélation, où chaque élément *(i, j)* décrit la similarité $Sim(q_i , t_j)$ entre un mots clés de la requête $q_i$ et les termes du document indexé $t_j$. A partir de cette matrice, un vecteur de similarité simVecrtor est généré. Tel que la longueur du vecteur est égale au nombre de mots-clés de la requête et chaque élément Vi du vecteur représente la *MAX(Sim(qi, tj))* de la ligne correspondante dans

la matrice. Cette étape garantit que seule la valeur de similarité la plus élevée est conservée pour les mots-clés de la requête avec un document donné. Enfin, afin d'obtenir le mapping document-requête « simDegree », nous procédons par le calcul de la similarité moyenne. Tel que [4] :

$$AVGSim(\text{SimVector}) = \frac{SUM(V_i)}{|SimVector|}$$




Dans notre cas, la similarité finale obtenue par ces calculs, est utilisée pour le remplissage du dataset, pour l'entrainement et les tests du Réseau de Neurones.

# 4 Conclusion

Au cours de ce chapitre, nous avons présenté l'architecture globale de notre proposition, en abordant en détails chaque partie et chaque composants de cette dernière, ainsi que le fonctionnement globale du Réseau de Neurone que nous avons utilisé pour notre solution de mapping et de ranking des documents pour une requête donnée.

Dans le prochain chapitre nous présenterons les outils de développement ainsi que les tests évaluation du système en utilisant différentes mesures.

# Chapitre IV :

# Tests et validations

# 1 Introduction

Après tout ce qui a été dit dans les chapitres précédents, et toutes les notions théoriques présentées, passons maintenant à l'étape d'implémentation et réalisation de notre système. Nous savons que pour aboutir à des similarités entre la requête et les documents il nous faut un réseau de neurones bien entrainé, cependant pour évaluer le degré de pertinence des correspondances obtenues, nous devons procéder à leur évaluation. Et ce, en utilisant quelques mesures comme : rappel, précision et f-mesure.

Dans ce chapitre nous parlerons du langage et outils de développement utilisés et nous présenterons également les résultats des tests d'évaluation de notre système.

# 2 Langage de programmation

Python est un langage de programmation open source, de haut niveau et multi-Platform, c.-à-d. qu'il est utilisable sur Linux, mac ou Windows, il connait depuis quelques années la plus grande croissance faisant de lui d'après le sondage au site Stack Overflow[15] le langage le plus populaire et c'est prévu de continuer comme ça dans les années à venir.

Python a passé par plusieurs version commençons par la version 0.9.0 qui a été publié la première fois en 1991, passons par la publication de python 2.0 en 2000, arrivons au Python 3.0 publier en 2008, et finissons par la dernière version 3.9 qui a été publier en Octobre 2020 [33] .

Python est maintenant utilisé dans nombreuses disciplines. Nous pouvons l'utiliser en développement Web. Il y'a même des Framework[16] très évolué comme « Django » qui sont utiles pour bien structurer le travail en Python. On le trouve aussi dans le Data Science et Data Analyse avec ses

[15] **Stack Overflow** est un site web proposant des questions et réponses sur un large choix de thèmes concernant la programmation informatique.

[16] **Framework** est une boite à outils qui contient des composants autonomes qui permettent de faciliter le développement d'un site web ou d'une application.

librairies (Pandas ,NumPy , ou Matlotlib). Python est également utilisé dans l'intelligence Artificielle et le Big Data. Les plus grandes entreprise du monde utilisent Python, Google par exemple l'utilise pour améliorer ces résultats de recherche et Netflix[17] pour les recommandations de ces clients.

# 3 Environnement de développement

Spyder est un acronyme signifiant «Scientific PYthon Development EnviRonment», c'est un IDE[18] open source. Le moyen le plus simple de démarrer avec Spyder est d'installer la distribution Anaconda. Anaconda est une distribution populaire pour la science des données et l'apprentissage automatique.

La distribution Anaconda comprend des centaines de packages, notamment NumPy, Pandas, scikit-learn, matplotlib, etc.

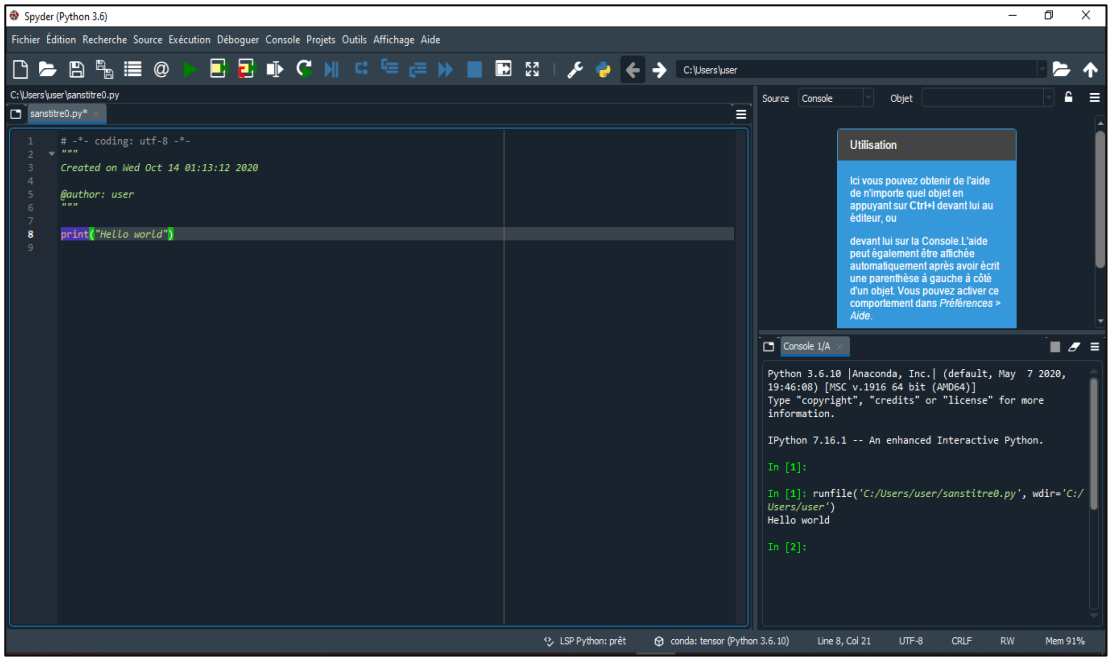



Figure 17 Environnement Spyder

[17] **Netflix** est une plateforme de vidéos à la demande par abonnement, qui permet d'accéder à plusieurs milliers de films, séries, dessins animés et documentaires, depuis n'importe quel terminal connecté à Internet.

[18] : Integrated Development Environment est une interface qui permet de développer, compiler et exécuter un programme dans un langage donné .

# 4 Outils et librairies utilisés

Les bibliothèques que nous avons utilisées pour notre travail sont :

- **Pandas** : elle permet la manipulation et l'analyse des données, nous l'avons utilisé pour lire le dataset.
- **Scikit-learn** : c'est la principale bibliothèque d'outils dédiés au machine learning et à la science des données dans l'univers Python, nous avons utilisé plusieurs modules de cette dernière pour la création de notre modèle, le training et les tests.
- **Pickle** : elle permet de sauvegarder une ou plusieurs variables dans un fichier et de récupérer leurs valeurs ultérieurement. Les variables peuvent être de type quelconque, nous l'avons utilisé pour sauvegarder le modèle des réseaux de neurones après le training.

# 5 Codes et IHM

Après avoir présenté tous les outils et l'environnent que nous avons utilisé pour développer notre système, passons maintenant à une vue plus proche et concrète. Dans cette section, nous présentons l'interface du système, et ses différentes fonctionnalités.

Lors de l'exécution, l'interface ci-dessous (figure 19) s'affichera, cette interface permet à l'utilisateur d'émettre sa requête, et choisir le type de calcul de similarité qui lui convient.


Figure 18 Interface de recherche

Après le choix de similarité, et en cliquant sur le bouton choisit, l'interface d'affichage apparait, avec les 3 premiers documents qui répondent le plus au besoin d'utilisateur, classés selon leur pertinence, comme illustré dans les deux figures qui suivent :

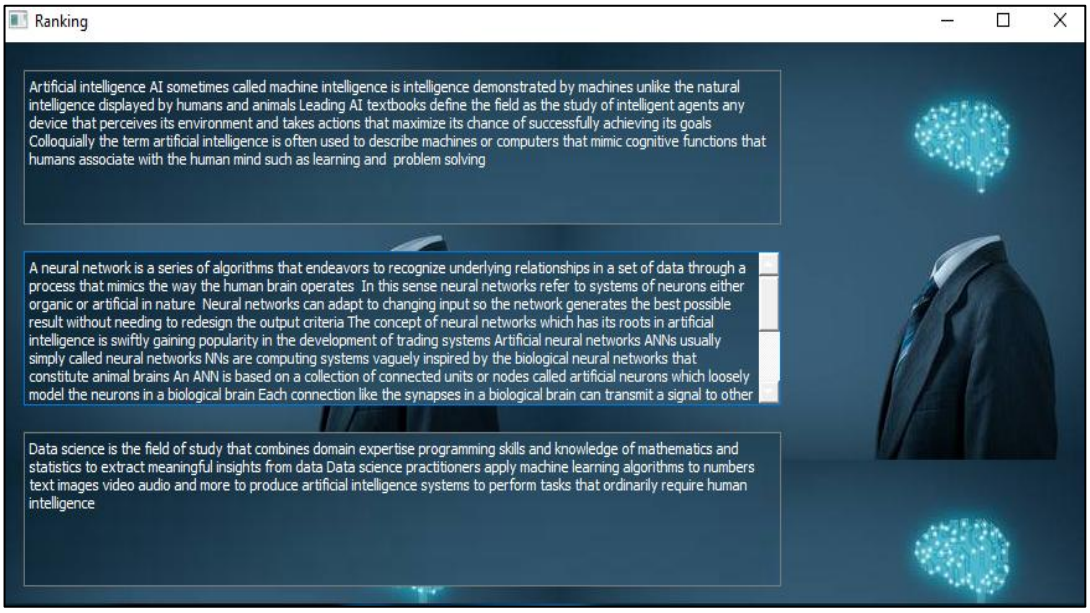



Figure 19 Résultat avec similarité cosinus.

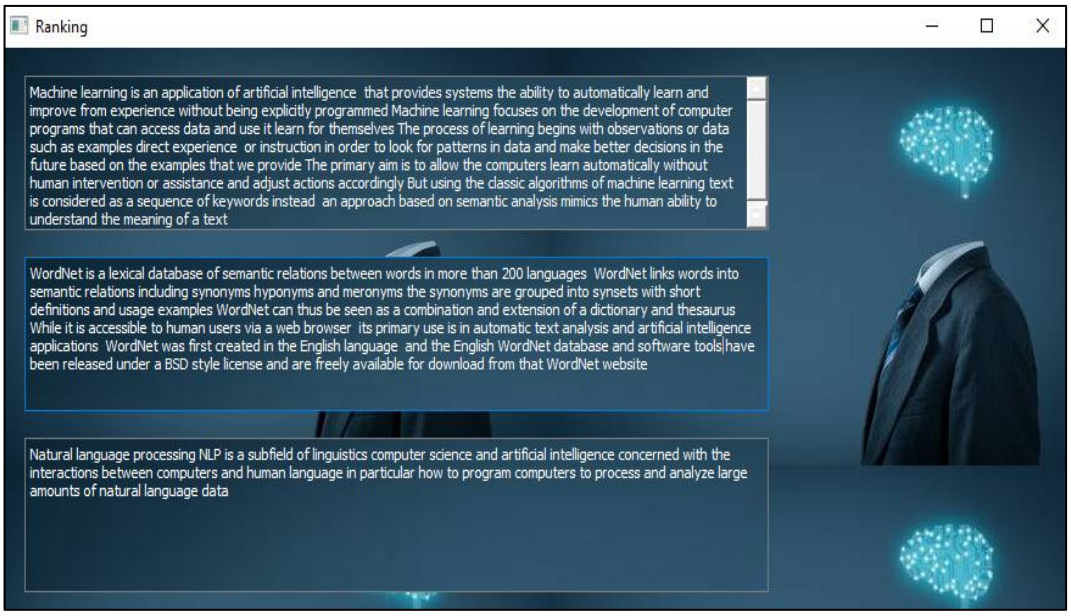



Figure 20 Résultat avec similarité Terminologique.

Après la présentation des interfaces graphique ; parlons maintenant ce qui se passe derrière.

Ces bouts de code ci-dessus, présentent comment nous avons créé le réseau de neurones, son entrainement, tests, évaluation et sauvegarde du meilleur modèle pour une utilisation ultérieure.

```
data = pandas.read_csv("Dataset-global.csv")


data_feature_vectors = []
for sample_idx in range(len(data)):
    document = [data.iloc[sample_idx]["Document"]]
    query = [data.iloc[sample_idx]["Query"]]
    feature_vector = extract_feature_vector(document, query)
    data_feature_vectors.append(feature_vector)

y = data["Similarity"]
X = data_feature_vectors

X_train, X_test, y_train, y_test = train_test_split(X, y,test_size=0.3,random_state=10)

regr = MLPRegressor(hidden_layer_sizes=(300, 150, 100, 50, 10), max_iter=50000)
regr.fit(X_train, y_train)

# Train
print(regr.predict(X))
# Test
print(regr.predict(X))

print("Train score : ", regr.score(X_train, y_train))
print("Test score : ", regr.score(X_test, y_test))

#to evaluate the model
print("r2 score: ",r2_score(y,regr.predict(X)))

# save the model to disk
filename = 'NNSimilarity1.pickle'
pickle.dump(regr, open(filename, 'wb'))

# load the model from disk
loaded_model = pickle.load(open(filename, 'rb'))
result = loaded_model.predict(X_test)
```

Figure 21 Création d'un réseau de neurone.

La figure suivante (figure 23) montre comment nous avons chargé le modèle neuronal pour le réutiliser, pour la sélection des document et le Ranking.

```
# Load the trained model.
filename = 'NNSimilarity1.pickle'
loaded_model = pickle.load(open(filename, 'rb'))

predicted_similarities = loaded_model.predict(X_pred)
# Indices of the best similarities in descending order. The first one is the best and the last one is the worst.
ranked_similarities_idx = sorted(range(data.shape[0]), key=lambda idx: predicted_similarities[idx], reverse=True)
print("Best Similarity index",ranked_similarities_idx)
# Order of the similarities in descending order. The first one is the best and the last one is the worst.
ranked_similarities = predicted_similarities[ranked_similarities_idx]

# Index of the best document.
best_document_idx = ranked_similarities_idx[0]
best_document_idx1 = ranked_similarities_idx[1]
best_document_idx2 = ranked_similarities_idx[2]
# The best document.
best_document = data.iloc[best_document_idx]["Document"]
best_document1 = data.iloc[best_document_idx1]["Document"]
best_document2 = data.iloc[best_document_idx2]["Document"]
# The similarity score of the best document.
best_document_similarity = ranked_similarities[0]

print()
#affichage ta3 index of the best doc
print("The index of the document of highest similarity : {sim_idx}".format(sim_idx=best_document_idx))
print("The index of the 2nd document of highest similarity : {sim_idx1}".format(sim_idx1=best_document_idx1))
print("The index of the 3rd document of highest similarity : {sim_idx2}".format(sim_idx2=best_document_idx2))
#affichage ta3 best doc
g=("{best_doc}".format(best_doc=best_document))
gg=("{best_doc1}".format(best_doc1=best_document1))
ggg=("{best_doc2}".format(best_doc2=best_document2))
```

Figure 22 Le Ranking

# 6 Mesure d'évaluation de performance

Après avoir entrainé et sauvegardé notre modèle neuronale, nous procédons dans cette section à l'évaluation, nous avons calculé les mesures de performance de chaque modèle notamment : rappel, précision et f-mesure.

Avant d'introduire les différentes mesures de calcules, abordons d'abord les notions suivantes (aussi illustré dans la figure 24) :

- **Vrai Positif** (VP) : c'est-à-dire le modèle indique un résultat positif alors que le fait étudié correspond à un cas positif (résultat de la réalité = résultat prédit).
- **Vrai Négatif** (VN) : c'est-à-dire le modèle indique un résultat négatif alors que le fait étudié correspond à un cas négatif (résultat de la réalité = résultat prédit).

- **Faux Positif** (FP) : c'est-à-dire le modèle indique un résultat positif alors que le fait étudié correspond à un cas négatif (résultat de la réalité ≠ résultat prédit).
- **Faux Négatif** (FN) : c'est-à-dire le modèle indique un résultat négatif alors que le fait étudié correspond à un cas positif (résultat de la réalité ≠ résultat prédit).

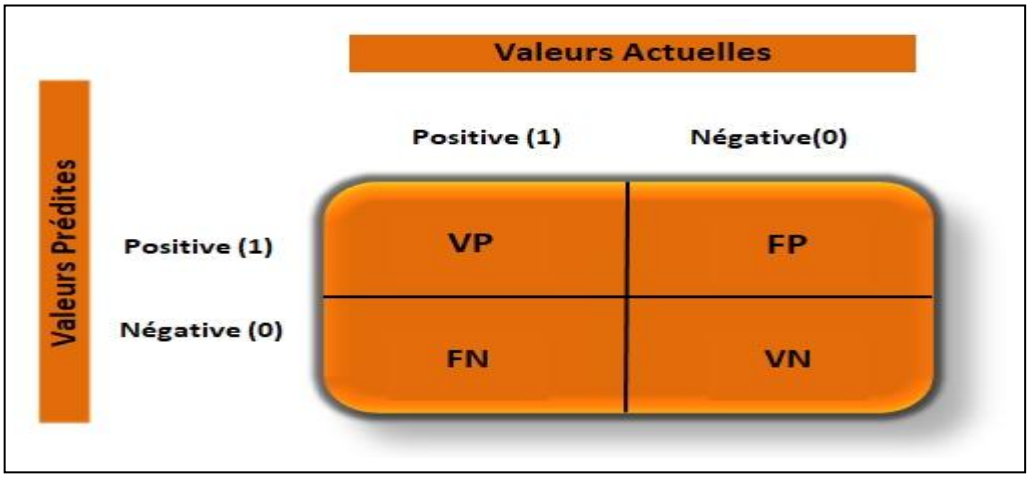



Figure 23 Matrice De Confusion[19]

### 6.1 Rappel

Le rappel permet de répondre à la question suivante : parmi toutes les étiquettes positives possibles, pour combien d'entre elles, le modèle a-t-il correctement identifié. Il est calculé comme suit :

$$R = \frac{\mathrm{VP}}{\mathrm{VP} + \mathrm{FN}}$$

### 6.2 Précision

La précision permet de répondre à la question suivante : quelle proportion d'identifications positives était effectivement correcte ? Elle est calculée comme suit :

[19] Une mesure des performances pour un problème de classification d'apprentissage automatique où la sortie peut être deux classes ou plus. C'est un tableau avec 4 combinaisons différentes de valeurs prévues et réelles.

$$P = \frac{\text{VP}}{\text{VP} + \text{FP}}$$

### 6.3 F-mesure

Moyenne harmonique de la précision et du rappel. Mesure la capacité du système à donner toutes les solutions pertinentes et à refuser les autres. Elle est calculée comme suit :

$$F = 2.\frac{(\text{P}.\text{R})}{(\text{P} + \text{R})}$$

Après le calcul de VP, VN, FP, FN, nous pouvons effectuer le calcul des mesures de performance, les tableaux suivants résument les résultats de calcul obtenus (respectivement les temps d'exécution et les mesure de performance).

| | Algorithme | Seconds | Minute |
|---|---|---|---|
| **Indexation des Documents** | **Secas** | 1.26 | 0.021 |
| **Indexation des Requêtes** | | 0.9 | 0.015 |

Tableau 3 Temps d'exéution d'indexation des requêtes et documents avec SECAS.

| | Modèle | Seconds | Minute |
|---|---|---|---|
| **Calcul des appariements documents-requêtes et Ranking** | **Cosinus** | 0.20 | 0.003 |
| | **Syntaxique** | 0.21 | 0.003 |
| | **Lexicale** | 0.11 | 0.001 |
| | **Terminologique** | 0.10 | 0.001 |
| | **Structurelle** | 0.13 | 0.002 |
| | **Sémantique** | 0.13 | 0.002 |

Tableau 4 : Temps d'exécution des calculs des appariements et du ranking.

| | Cosinus | Syntaxique | Lexicale | Terminologique | Structurelle | Sémantique |
|---|---|---|---|---|---|---|
| **Rappel** | 0,94 | 0.99 | 0.56 | 0.85 | 0.76 | 0.81 |
| **Précision** | 1 | 0.70 | 0.86 | 1 | 1 | 1 |
| **F-Mesure** | 0.96 | 0.82 | 0.68 | 0.92 | 0.86 | 0.90 |

Tableau 5 : Mesures de performance de chaque modèle neuronal.

Pour une meilleure lisibilité et visualisation des résultats, nous les avons illustrés grâce à l'histogramme suivant :

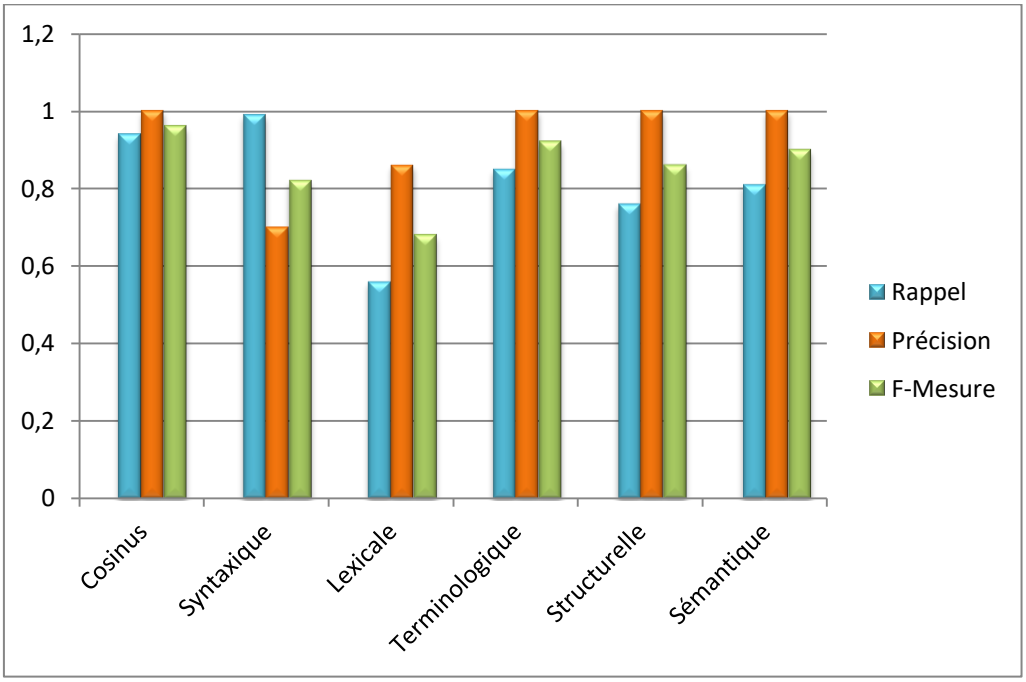



Figure 24 Mesure de performance de chaque modèle

# 7 Interprétation des résultats

A travers les tableaux et l'histogramme présentés précédemment, nous pouvons clairement constater que de bons résultats ont été obtenus par notre système. Parlons du temps d'exécution : le temps a été diminué à cause d'utilisation du Réseau de Neurone qui nous a permis de faire un traitement (indexation, calcule de similarité et classement) sur tous les documents en même temps.

Concernant les mesures de performance, nous avons obtenu des résultats encourageants avec un seuil mis à 0.7 pour chaque similarité. Prenons comme exemple le modèle structurel qui a eu un score de précision de 1. Ce qui signifie que les résultats sont à 100% précis, avec un rappel de 0.76 ce qui veut dire qu'il a trouvé 76% des réponses pertinentes possibles, de même le modèle Terminologique a eu un score de précision de 1 avec un

score de rappel de 0.85 ce qui veut dire qu'il a trouvé 85% des documents possibles.

# 8 Conclusion

Dans ce chapitre nous avons décrit les outils et l'environnement utilisé pour le développement de notre système. Nous avons aussi présenté l'interface graphique et des extraits de code illustratifs.

Nous avons fait également l'évaluation du système en utilisant différentes mesure de performance (Rappel, Précision, et F-mesure ), et nous avons clôturé le chapitre par l'interprétation des résultats obtenus.

# Conclusion Générale

# *Conclusion Générale et Perspective*

La Recherche d'Information (RI) est définie comme une activité dont la finalité est de localiser et de délivrer un ensemble de documents à un utilisateur en fonction de son besoin en information. Le défi est de pouvoir, parmi le volume important de documents disponibles, trouver ceux qui correspondent au mieux à l'attente de l'utilisateur.

Dans le cadre de ce projet, nous avons essayé de donner une nouvelle perspective on ce qui concerne l'appariement requête-document en utilisant le concept des Réseaux de Neurones.

Premièrement, nous avons utilisé l'algorithme « SECAS » pour produire de bons Stems qui nous a conduit à une bonne représentation des documents, qui à son tours nous a mené vers un calcul de score de similarité convenable. Le tout pour retourner des documents pertinents dans un bon classement afin de satisfaire le besoin en Information de l'Utilisateur.

Cependant, nous constatons que la notion de pertinence dépend de la satisfaction de l'utilisateur d'une part, et des différents sens portés par les termes de la requête d'une autre part, et c'est ici qu'interviennent les différentes mesures de similarité. Ces mesures nous ont permis d'effectuer un appariement à base de différents critères (le sens, la structure du terme ...etc.).

Deuxièmement, l'intégration du concept de Réseau de Neurone qui nous a donné des résultats encourageants concernant le temps d'exécution, car il nous a permis de calculer l'appariement requête-document de manière parallèle. C'est-à-dire la similarité d'une requête est calculée simultanément avec tous les documents de la collection.

Les résultats obtenus sont bons, et le système a montré une bonne performance, mais cela ne nous empêche pas d'envisager quelques perspectives qui permettent la l'amélioration de notre travail notamment :

- L'utilisation de datasets plus grands pour effectuer des tests plus intensifs et de sortir avec de nouvelles perspectives.
- Entrainer le Réseau de Neurone sur des ressources linguistiques autres que l'Anglais. Par exemple en langue Arabe qui est au cœur des recherches d'actualité en TAL.
- Ça sera aussi intéressant d'envisager l'intégration des dimensions contextuelles (le temps, la géolocalisation, l'environnement social…) dans le processus d'appariement afin de le rendre plus dynamique et de pouvoir répondre au mieux aux requêtes des utilisateurs.

"Every Beginning Has an End.

But Every End is the Start of a New Beginning ..."

# Références bibliographiques

[1] M. S. e. A. A. Habiba, *Tests et analyse d'algorithmes d'indexation sémantique dans le cadre de la proposition d'un système de recherche d'information sensible au contexte,* Octobre 2019.

[2] V. Bush, «AS_WE_MAY_THINK,» *Atlantic Monthly,* 1945.

[3] Y. S. A. e. S. Gupta, *A new fuzzy logic based ranking function for efficient information retrieval system. Expert Systems with Applications,* 2015.

[4] G. Salton, Automatic information organization and retrieval, McGraw Hill Text, 1968.

[5] Ingwersen, «elements of a cognitive theory for information retrieval interaction,» chez *Polyrepresentation of information needs and semantic entities*, New York, NY, USA, 1994, p. 101–110.

[6] Smeaton, A. F, chez *Natural language processing and information retrieval.*, Inf. Process. Manage, 1990, p. 19–20.

[7] A. Meftah, *Un modèle de reformulation des requêtes pour la recherche d'information sur le Web.*

[8] T. L. L. G. a. S. D. George Furnas, «The vocabulary problem in human-system communication.,» *Communications of the ACM,* p. 30(11):964–971, 1987.

[9] H. P. Luhn, «A statistical approach to mechanized encoding and

searching of literary information,» *IBM Journal of research and development*, p. 1(4):309–317, 1957.

[10] M. E. M. a. J. L. Kuhns, «On relevance, probabilistic indexing and information retrieval,» *ACM (JACM)*, p. 7(3):216–244, 1960.

[11] M. S. Mohammad, *Indexation automatique et la Recherche D'information dans les documents,* 2006.

[12] W Bruce Croft, Donald Metzler, and Trevor Strohman, Search engines: Information retrieval in practice, Addison-Wesley Reading, 2015.

[13] M. MEZZI, *CONTEXT-AWARE INFORMATION RETRIEVAL SYSTEMS: CONTRIBUTION TO A SEMANTICALLY ENRICHED, FOLKSONOMY-BASED TEXT-SEARCH.*, June 2018.

[14] C. Tambellini, *"Un système de recherche d'information adapté aux donnée,incertaines: adaptation du modèle de langue". Thèse de doctorat en informatique,Université de Nice-Sophia Antipolis-UFR sciences*, 2007.

[15] N. J. B. a. W. B. Croft, «Information filtering and information retrieval: Two sides of the same coin?,» *Communications of the ACM, 35(12)*, p. 29–38, December 1992.

[16] D. Harman, «Towards interactive query expansion,» *In Proceedings of the 11th annual international ACM SIGIR conference on Research and development in information retrieval*, p. 321–331, 1988.

[17] S. E. Robertson, «On term selection for query expansion,» *Journal of documentation, 46(4),* p. 359–364, 1990.

[18] C. v. Rijsbergen, *Information Retrieval,* second edition ,London: Butterworth & Co (Publishers) Ltd, 1979.

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

[20] S. R. a. K. S. Jones, «Relevance weighting for search terms.,» *Journal of The American Society for Information Science,* pp. 129-146, 1976.

[21] G. S. a. M. McGill, *Introduction to Modern Information Retrieval.,* New York: McGraw-Hill, 1986.

[22] K. Garrouch, *Modèles de Recherche d'information basés sur les Réseaux Bayésiens et les Réseaux Possibilistes,* 2017.

[23] Hermann, «OverBlog,» 06 Décembre 2014. [En ligne]. Available: http://profsecretariat.over-blog.com/2014/12/organisation-et-methodes-administratives.html#:~:text=Le%20classement%20est%20le%20rangement,agenc%C3%A9s%20et%20ayant%20un%20sens.. [Accès le 17 Octobre 2020].

[24] J.-P. Lardy, *Méthodes de tri des résultats des moteurs de recherche,* 31 Mai 2020.

[25] S. K. Robertson S. E., «"Relevance weighting of search terms",» *Journal of the American society for Information Science,* p. 27, 1976.

[26] E. Negre, *Comparaison de textes: quelques approches,* Octobre 2013.

[27] S. a. H. G. Mohammad, *Distributional measures of semantic distance :A survey. CoRR, abs/1203.1858.,* 2012.

[28] A. Montazeralghaem et R. R. a. J. Allan, *Relevance Ranking Based on Query-Aware Context Analysis,* 2020.

[29] B. L, «Le Big Data,» 05 Avril 2019. [En ligne]. Available: https://www.lebigdata.fr/reseau-de-neurones-artificiels-definition#:~:text=Par%20exemple%2C%20un%20r%C3%A9seau%20de,ordinateur%20%C3%A0%20reconna%C3%AEtre%20des%20objets.&text=Le%20r%C3%A9seau%20de%20neurones%20analyse,la%20pr%C3%A9cision%20de%20l'algorit. [Accès le 17 Octobre 2020].

[30] Rod, «MonCoachData-Comprendre les Réseaux de Neurones,» 05 Avril 2019. [En ligne]. Available: https://moncoachdata.com/blog/comprendre-les-reseaux-de-neurones/#:~:text=Les%20r%C3%A9seaux%20de%20neurones%20sont,les%20r%C3%A9sultats%20(si%20n%C3%A9cessaire).. [Accès le 17 Octobre 2020].

[31] [En ligne]. Available: https://www.lemagit.fr/definition/Reseaux-de-neurones-recurrents. [Accès le 12 Novembre 2020].

[32] «Open Class Rooms,» [En ligne]. Available: https://openclassrooms.com/fr/courses/4470531-classez-et-segmentez-des-donnees-visuelles/5082166-quest-ce-quun-reseau-de-neurones-convolutif-ou-cnn. [Accès le 12 Novembre 2020].

[33] «Python Documentation,» [En ligne]. Available: https://docs.python.org/3/. [Accès le 15 Octobre 2020].

[34] C. Y. a. W. M. [] F. Liu, « IEEE Transactions on Knowledge and Data Engineering, 16(1),» chez *Personalized web search for improving retrieval effectiveness.*, 2004, p. 28–40.