

République Algérienne démocratique et populaire

Ministère de l'enseignement supérieur de la recherche scientifique

Université Chadli Bendjedid

Faculté des sciences et de la technologie

Département d'informatique



UNIVERSITE CHADLI BENDJEDID - ELTARF

الجمهورية الجزائرية الديمقراطية الشعبية

وزارة التعليم العالي والبحث العلمي

جامعة الشاذلي بن جديد

كلية العلوم والتكنولوجيا

قسم الاعلام الآلي

MEMOIRE

Présenté par
Kabouche Khadidja

Pour l'obtention de diplôme de

MASTER

Filière : Informatique

Spécialité : Systèmes Informatiques Intelligents

Thème

"Conception et réalisation d'un système de recherche
d'information basé sur l'apprentissage par
renforcement profond "

Soutenue le : 27/ 10 /2020

Devant le Jury composé de :

Qualité	Nom et Prénom	Grade	Université
Présidente	Mme. Makhlouf Amina	MCB	Chadli Bendjedid El-Tarf
Rapporteur	Mme.Maatalah Majda	MCB	Chadli Bendjedid El-Tarf
Examinatrice	Mme.Bougerne Imene	MCB	Chadli Bendjedid El-Tarf

Année Universitaire : 2019/2020

Remerciements

Nous remercions, en premier lieu, notre dieu qui a bien voulu nous donner la force pour effectuer le présent travail.

Nous tenons à exprimer nos sincères gratitudes et remerciement à nos professeurs qui nous ont apportés de l'aide et qui ont contribué à l'élaboration de ce rapport de stage :

Tout d'abord :

nous tenons à exprimer, nos vifs remerciements et notre gratitude à ma promotrice Dr. « Maatalah Majda » Docteur à l'université Chadli Bendjedid elTaref. Pour sa précieuse aide et son suivi permanent durant toute la période du projet.

Nos sincères remerciements vont également aux membres du jury pour avoir accepté de juger notre modeste travail.

Nous remercions « Makhlouf Amina », d'avoir si complaisamment accepté de participer à notre jury et d'en être le Président.

Nous sommes reconnaissants envers « Bougerne Imene » d'avoir bien voulu étudier ce travail et participer à la commission d'examen afin de le juger.

En fin à tous ceux qui nous ont encouragés de près ou de loin à l'élaboration de ce travail de recherche.

Dédicace

Je dédie ce mémoire

A mon père

Pour son grand amour, sa patience, son encouragement, son sens du devoir et ses sacrifices pour que je réussisse dans mes études.

A ma mère

Pour son affection, sa patience, son encouragement pendant les épreuves difficiles ainsi que ses prières qui m'apportent bonheur et réussite.

A mon frère : « Abdel malek » Pour son soutien et ses encouragements.

A mon mari : « Raouf » Je te remercie du fond du cœur, Ton soutien me donne la force de continuer

A mes meilleurs amies

A toute les étudiants de master promotion informatique 2020

A tous mes proches et famille

« magri » et « kabouche » et « Nouri »

Table des matières

Remerciements

Dédicace

Table des matières

Liste des figures

Liste des tableaux

Liste des acronymes

Introduction générale

Chapitre 1: la recherche d'information et l'apprentissage par renforcement profond 1

Partie 1: La Recherche d'Information 1

1.1. Introduction..... 1

1.2. Définition de la recherche d'information..... 2

1.3. Problématique de la recherche d'information..... 2

1.4. Historique de la Recherche d'information..... 3

1.5. Notions de base de le RI.....4

1.6. Processus de le RI.....5

1.6.1. L'indexation..... 5

a) Définition..... 6

b) Les types d'indexation 6

c) Résultats d'indexation..... 7

d) Les langages de l'indexation 7

e) Processus d'indexation..... 8

1.6.2. L'appariement document-requête 10

1.6.3. La reformulation de requête..... 11

❖ Les approches pour la reformulation de requêtes 11

1.7. Les systèmes de recherche d'information.....	12
1.7.1. Définition.....	12
1.7.2. Modélisation du système de recherche d'information	13
❖ Modèle booléen.....	14
❖ Modèle Probabiliste.....	15
❖ Modèle Vectoriel.....	15
1.7.3. Evaluation des systèmes de recherche d'information	16
1.8. Domaines d'application de la RI	18
Partie 2: l'apprentissage par renforcement profond	18
1.9. Concepts connexes	19
1.9.1. L'intelligence artificielle (IA).....	19
1.9.2. L'apprentissage automatique	19
1.9.3. L'apprentissage en profondeur	19
1.9.4. L'Apprentissage par renforcement	20
❖ Définition.....	20
❖ Modélisation avec l'apprentissage par renforcement	20
❖ Exemples d'utilisation de l'apprentissage par Renforcement.....	21
1.9.5. L'apprentissage par renforcement profond.....	21
❖ Définition.....	21
❖ Les algorithmes d'apprentissage par renforcement profond.....	22
❖ Quelques modèles d'apprentissage par renforcement profond	24
❖ Une approche d'apprentissage par renforcement profond	25
Partie 3: La recherche d'information et l'apprentissage par renforcement profond.....	25
1.10.1 .travaux connexes	26
1.10.2. Optimisation des évaluations de requêtes à l'aide de l'apprentissage par renforcement pour la recherche sur le Web	27
1.10.3. Apprentissage par renforcement profond pour la recherche, la recommandation et la publicité en ligne: une enquête	27

1.10.4. Diriger la recherche exploratoire: apprentissage par renforcement des interactions des utilisateurs avec des mots clés	28
1.10.5. Utiliser l'apprentissage par renforcement pour connaître le classement de pertinence des requêtes de recherche.....	29
1.11. Conclusion.....	30
Chapitre 2 : conception d'un système de recherche d'information.....	31
2. Introduction	31
2.1. Problématique	31
2.2. Objectif et motivation.....	32
2.3. Un système de recherche d'information	33
2.3.1. Architecture du système	34
2.3.2. La description du système.....	35
2.3.2.1. La représentation du monde	35
2.3.2.2. Pré-traitement.....	36
a) Préparation de données	36
b) La tokenization	36
c) Elimination des mots vides.....	36
d) Lemmatisation ou racinisation.....	36
2.1.1.1. Algorithmes utilisés.....	37
a) Perceptron multicouche (multilayer perceptron MLP).....	37
➤ La fonction Coût.....	39
➤ Architecture du réseau	39
➤ Etapes de l'algorithme	39
2.2. Conclusion	40
Chapitre3 : Implémentation et évaluation	41
3. Introduction	41
3.1. Dataset	41
3.2. Présentation des outils de développement	43
3.2.1. Matériel utilisée	43

3.2.2. Logiciels et Plateformes utilisés	44
3.3. Objective des expérimentations	45
3.3.1. Métriques d'évaluation	45
3.3.2. Méthodologie d'implémentation.....	46
1.4. Expérimentations et Tests	48
1.4.1. Expérimentation 1 : (Couches Cachées = 8).....	48
1.4.2. Expérimentation 2 : Couches Cachées = 16.....	50
1.4.3. Expérimentation 3 : Couches Cachées = 24.....	52
Partie 2 : Obtenez une prédiction pour la maladie.....	55
1.5. Conclusion	57

Conclusion et perspectives

Références

A. Références Bibliographiques

B. Références Web

Résumé

Liste des figures

Figure 1.1 : Problématique de la recherche d'information	3
Figure 1.2 : Processus de RI.....	5
Figure 1.3 : Les étapes du processus d'indexation	8
Figure 1.4 : Définition d'un système de recherche d'information. Error! Bookmark not defined. 12	
Figure 1.5 : Processus de RI et le système de recherche d'information. Error! Bookmark not defined. 13	
Figure 1.6 : Taxonomie des modèles de RI proposés.....	14
Figure 1.7 : Similarité selon le modèle vectoriel	16
Figure 1.8 : Modélisation d'un agent d'apprentissage par renforcement	21
Figure 2.1 : Architecture globale du système de recherche d'information	34
Figure 2.2 : Les couches de MLP.....	39
Figure 3.1 : Exemple de données utilisées.....	43
Figure 3.2 : courbe de loss avec couche cachée =8	49
Figure 3.3 : présente les résultats du Rappel et de la précision	50
Figure 3.4 : Courbe de Rappel et Précision avec couche cachée =8... ..	50
Figure3.4 : courbe de la fonction de perte avec 16 couches cachées.....	51
Figure3.5 : Courbe de Rappel et Précision avec 16 couches cachées.....	52
Figure 3.6 : Courbe de la fonction de perte avec 24 couches cachées.....	53
Figure 3.7 : Courbes de Rappel et Précision avec 24 couches cachées	54
Figure 3.8 : résultat de prédiction avec requête 1.....	55
Figure 3.9 : résultat de prédiction avec requête 2.....	56
Figure 3.10 : résultat de prédiction avec requête 3.....	56
Figure 3.11 : résultat de prédiction avec requête 4.....	57

Liste des tableaux

Tableau 1.1: La table de contingence

Tableau 3.1 : nombre de documents de chaque classe dans la base de données

Liste des acronymes

SRI	Systèmes de Recherche d'Information
RI	recherche d'information
IA	l'intelligence artificielle
TF	Terme Frequency
IDF	Inverse document frequency
RSV	Retrieval Status Value
RL	Reinforcement learning
MLP	Perceptron multicouches
CNN	Réseaux de neurones convolutionnels
RNN	Réseaux de neurones récurrents
MDP	Processus de décision de Markov
DRL	Deep Reinforcement Learning
DQN	Réseau Q Profond
NLP	Traitement du langage naturel
WAIR	Web-Agents for Information Retrieval
CTR	Click-through Rate
ROI	Return on Investment
GD	Guaranteed Delivery
RTB	Real-Time Bidding
MARL	Multi-Agent Reinforcement Learning
TREC	Text REtrieval Conference
RLR	Reinforcement Learning Retrieval
DL	deep learning

Chapitre 1 : la recherche d'information et l'apprentissage par renforcement profond

Le chapitre présent est divisé en deux grandes parties, où la première est consacrée pour la présentation du domaine de la recherche d'information et la deuxième pour un état de l'art sur l'utilisation des algorithmes d'apprentissage en profondeur dans la recherche d'information.

Partie 1 : La Recherche d'Information

1.1. Introduction

Au début des années soixante, quelques années après l'invention de l'ordinateur, la recherche de l'information est apparue. C'est la science de la recherche de l'information dans des documents (dans les documents eux-mêmes, dans les méta-données qui décrivent les documents ou encore, dans les relations qu'entretiennent les documents entre eux), qu'ils soient dans une base de données, dans une base documentaire ou sur le Web.

Le monde de la recherche s'est intéressé aux Systèmes de Recherche d'Information (SRI) qui sont les outils dans lesquels sont mis en œuvre des techniques et des mécanismes assurant la gestion automatique des informations documentaires.

La représentation interne d'un document est généralement constituée d'un ensemble de termes indexés associés à des poids, c'est-à-dire à une valeur reflétant l'importance du terme dans le document dans lequel il apparaît. Cet ensemble est construit lors de l'opération d'indexation.

La recherche d'information est un domaine assez vaste qui contient plusieurs notions. Dans cette partie, nous allons définir brièvement les notions de base de ce domaine, en mettant l'accent sur la notion d'indexation et les modèles les plus utilisés par les systèmes de RI ainsi que les métriques de leur évaluation. [1]

1.2. Définition de la recherche d'information

Il existe différentes définitions pour donner du sens à la recherche d'information, nous mentionnons les deux définitions suivantes:

Définition 1 : « La recherche d'information (RI) est une branche de l'informatique qui s'intéresse à l'acquisition, l'organisation, le stockage, la recherche et la sélection d'information » [salton, 1968][2]

Définition 2 : « La RI est un domaine qui étudie la manière de répondre pertinemment à une requête utilisateur en retrouvant les informations qui correspondent aux mieux à ses attentes. Naturellement, les besoins en information varient d'un utilisateur à l'autre et peuvent également changer et/ou évoluer chez le même utilisateur d'un moment à l'autre.

Ainsi, afin de bien assister les utilisateurs dans leur recherche, les systèmes de recherche d'information (SRIs) doivent d'une part être capables d'identifier le besoin en information de l'utilisateur et de s'adapter à ses différents intérêts et préférences ainsi qu'à leur évolution dans le temps, et d'une autre part de repérer les requêtes équivalentes qui correspondent à un même besoin d'information. Celles-ci peuvent être exprimées différemment par le même utilisateur ou par différents utilisateurs. Cette adaptation reste un grand défi essentiel et ouvert pour les SRIs». [3]

1.3. Problématique de la recherche d'information

Quel que soit le type de documents que l'utilisateur interroge, il est rarement intéressé par la structuration de documents, car il souhaite accéder à des documents qui répondent à son besoin d'information par le contenu, ces informations doivent être suffisantes pour répondre à la demande que l'utilisateur écrit, qui consiste généralement en de simples mots clés, Pour que l'IRS puisse se concentrer sur les informations pertinentes du document, et la réponse qu'il fournit n'est que plus précise.[4]

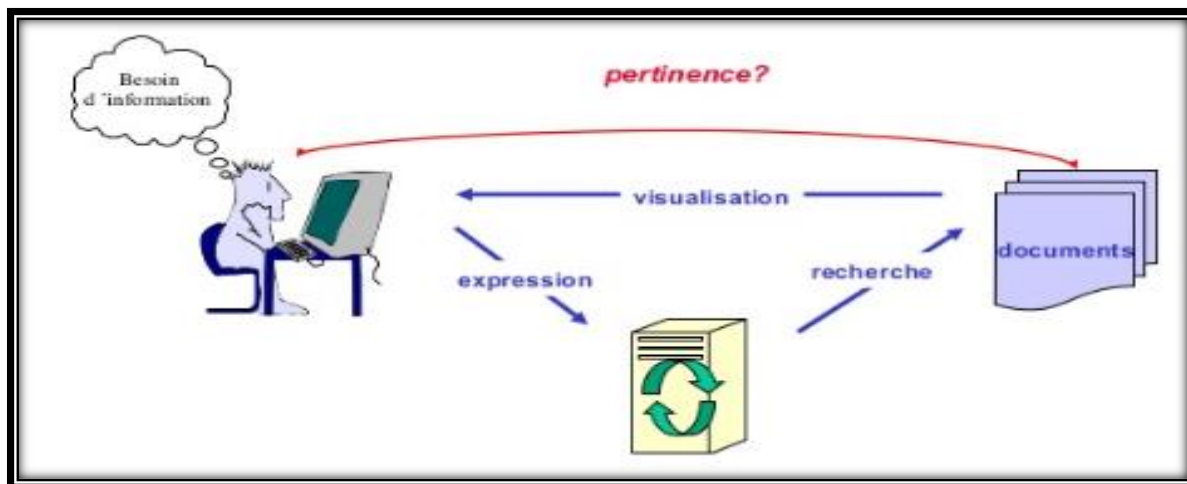


Figure 1.1: Problématique de la recherche d'information [w1]

1.4. Historique de la Recherche d'information

La RI n'est pas un domaine récent. Il date des années 1940, dès la naissance des ordinateurs. Au début, la RI se concentrait sur les applications dans des bibliothèques, d'où aussi le nom "automatisation de bibliothèques". Depuis le début de ces études, la notion de pertinence a toujours été un objet.

Dans les années 1950, on commençait de petites expérimentations en utilisant des petites collections de documents (références bibliographiques). Le modèle utilisé est le modèle booléen. Dans les années 1960 et 1970, des expérimentations plus larges ont été menées, et on a développé une méthodologie d'évaluation du système qui est aussi utilisée maintenant dans d'autres domaines.

Des corpus de test (e.g. CACM) ont été conçus pour évaluer des systèmes différents. Ces corpus de test ont beaucoup contribué à l'avancement de la RI, car on pouvait les utiliser pour comparer les différentes techniques, et de mesurer leurs impacts en pratique.

Le système qui a le plus influencé le domaine est sans aucun doute SMART, développé à la fin des années 1960 et au début des années 1970 par une équipe dirigée par le professeur G. Salton. Certaines techniques ont été implantées et expérimentées pour la première fois dans ce système, par exemple, le modèle vectoriel, la technique du retour de pertinence, et des développements sur le modèle probabiliste.

Les années 1980 ont été influencées par le développement de l'intelligence artificielle. Ainsi, on tentait d'intégrer des techniques de l'IA en RI, par exemple, le système expert pour la RI, etc.

Les années 1990 (surtout à partir du 1995) sont les années de l'Internet. Cela a pour effet de propulser la RI en avant scène de beaucoup d'applications. C'est probablement grâce à cela que vous entendez parler de la RI. La venue de l'Internet a aussi modifié la RI. La problématique est élargie. Par exemple, on traite maintenant plus souvent des documents multimédia qu'avant. Cependant, les techniques de base utilisées dans les moteurs de recherche sur le web restent identiques. [5]

1.5. Notions de base de le RI :

- **Besoin en information:** est une représentation mentale de ce que l'utilisateur souhaite rechercher, ce besoin est représenté sous forme de requête. [6]
- **Requête:** Une requête exprime le besoin d'information d'un utilisateur, elle représente l'interface entre le système de recherche d'information (SRI) et l'utilisateur. [5]
- **Document :** c'est toute unité qui peut constituer une réponse à une requête utilisateur, ça peut être un texte, un morceau de texte, une page Web, une image, une bande vidéo, etc.
- **Corpus :** est un ensemble d'informations exploitables et accessible par un utilisateur, ou c'est l'ensemble de documents dans lequel l'utilisateur cherche une information.

Pertinence : c'est le critère le plus important dans la RI et l'objet des systèmes de recherche d'information, cette notion n'est pas facile à définir car elle fait intervenir plusieurs notions, mais la définition la plus simple est que: "*la pertinence est la correspondance entre un document et une requête ou encore un degré de relation du document à la requête et une mesure d'utilité du document pour l'utilisateur*". [6]

La pertinence est liée à deux niveaux:

- Niveau système: le système mesure le degré de pertinence comme une valeur de similitude entre un document et une requête.

- Niveau utilisateur: pour l'utilisateur la pertinence correspond à la satisfaction d'ensemble du document restitué par le SRI. Deux utilisateurs peuvent juger différemment un même document envoyé pour une même requête. [6]

1.6. Processus de le RI :

Nous présentons dans cette section une figure explicative, (La Figure 2), qui schématise le processus de la recherche d'information (*Figure 1.2*), qui porte sur plusieurs étapes. [7]

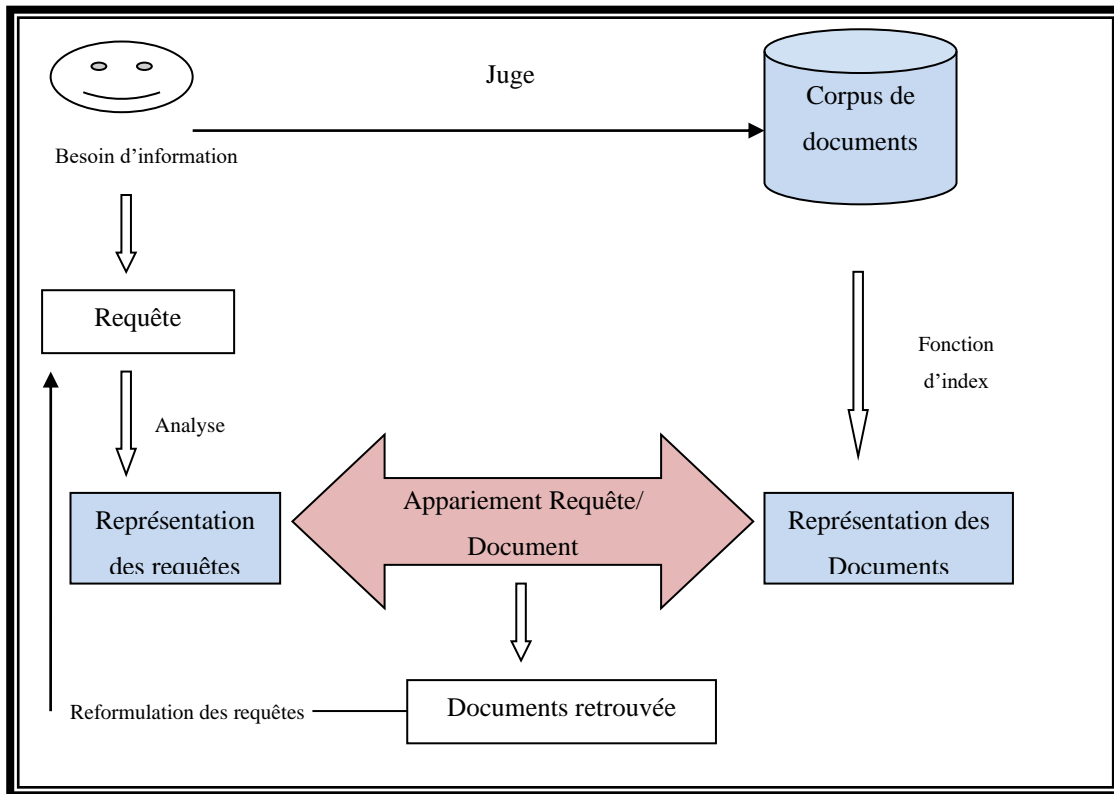


Figure 1.2 : processus de RI [w2]

6.1.1 L'indexation

Les concepts et méthodes de l'indexation se sont principalement constitués dans un contexte où les documents textuels, sur des supports de type papier, sont prédominants. Sous l'effet de la numérisation des contenus et de l'informatisation de leur exploitation, le cadre conceptuel et méthodologique

doit évoluer profondément pour affronter des problèmes inédits et tirer partie des nouvelles possibilités offertes par le cadre technologique émergent. [8]

a) Définition

« L'indexation est le passage d'un document textuel (ou une requête) à une représentation exploitable par un modèle de RI par la construction de mots clés appelé langage d'indexation. à son exploitation dans une application donnée (par exemple la recherche)».

Document textuel (ou requête) \longrightarrow exploitable par le SRI.
Indexation

Dans le contexte du numérique :

- Les documents sont destinés à être interprétés par des êtres humains.
- Les index sont des représentations du contenu des documents manipulés par des ordinateurs. [9]

Elle conduit à élaborer des outils de recherche documentaire : index, catalogues, bulletins bibliographiques, fichiers manuels ou automatisés, qui seront ensuite consultés et qui permettront la sélection de documents répondant à une question. [10]

b) Les types d'indexation

Il existe deux principales manières pour effectuer l'indexation :

- ✓ **manuelle** : L'indexation manuelle se fait par des experts en linguistique. Les experts analysent les textes et choisissent les termes descripteurs qui permettent de mieux représenter le contenu des documents. Le problème de cette indexation c'est que les index ne sont pas toujours les mêmes (deux experts différents peuvent choisir des descripteurs différents selon leurs visions).
- ✓ **automatique** : L'indexation automatique quant à elle permet de pallier aux problèmes liés aux interprétations différentes et permet donc de limiter les problèmes de disparité. L'indexation automatique est de deux types :
 - *Indexation orientée document* : Son objectif est de résumer ou de présenter le contenu de chaque document.
 - *Indexation orientée requête* : Pour chaque document, refléter les requêtes pour lesquelles il est pertinent : l'indexation d'un document doit alors représenter les raisons pour lesquelles un utilisateur

consulte ce document (i.e. : confronter chaque document de la base à une liste de requêtes prédéfinies).

Afin de satisfaire les besoins des utilisateurs différents, le système doit disposer d'un minimum d'informations par rapport aux documents. Dans ce cas il s'agira d'une indexation orientée requête. Afin de satisfaire le besoin d'un utilisateur donné l'indexation doit être orientée requête. [7]

c) Résultats d'indexation

Le résultat d'une indexation, appelé *l'index*, donne un ensemble de termes et leurs pondérations pour chaque document comme suit :

$$D_j \rightarrow \{ \dots (t_i, a_{ij}) \dots \} \quad (1.1)$$

Avec t le terme d'indice i dans le vocabulaire et a_{ij} son poids dans le document d_j . Cette représentation facilite la recherche des termes inclus dans un document.

Cependant, étant donné une requête contenant quelques termes, il faut rechercher les documents correspondant à chacun de ces termes. Pour cela *un fichier inversé* est construit avec la structure suivante :

$$t_i \rightarrow \{ \dots (d_j, a_{ij}) \dots \} \quad (1.2)$$

.[1]

d) Les langages de l'indexation

On peut distinguer deux types de langage d'indexation :

1) **Langage contrôlé** : Le langage contrôlé est construit à priori, et permet de limiter le nombre de représentations d'un document ou d'une requête, et il est régi par certaines règles afin d'éviter les problèmes de polysémie, synonymie, homonymie ... etc. dont on peut distinguer :

- ✓ **Le vocabulaire contrôlé** : Le vocabulaire contrôlé est conçu afin d'éviter toute ambiguïté, donc tous ses termes doivent être non ambigus et non redondants.
- ✓ **La Taxonomie**: est une collection de vocabulaires contrôlés auxquels une notion de hiérarchie est ajoutée. Chaque terme d'une taxonomie est relié aux autres par une relation de parent-fils.

- ✓ **Le Thésaurus** : est un réseau de vocabulaires contrôlés. Il s'agit d'une représentation de données incluant les relations associatives en plus des relations hiérarchiques.
- ✓ **L'Ontologie** : Est un vocabulaire contrôlé exprimé dans un langage de représentation. Ce langage a une grammaire d'utilisation des termes du vocabulaire dans un domaine donné
Par rapport aux Thésaurus, une ontologie inclut les instances et elle permet de définir les attributs de ces instances.

2) **Le langage libre** : est proche de langage naturel des humains, il n'est pas contrôlé, ni régit par aucune syntaxe. Son vocabulaire et ses éléments sont choisis à postériori. Son vocabulaire évolue rapidement et peut contenir des termes synonymes, polysémiques ...etc. Par conséquent, des documents portants sur le même sujet peuvent être représentés par des descripteurs différents, et inversement (sujets différents et mêmes descripteurs choisis). Le problème se pose au niveau des requêtes, car un document peut être choisi comme répondant au besoin définit par la requête uniquement parce que leurs descripteurs sont les mêmes. [7]

e) Processus d'indexation

La (Figure 1.3) représente les étapes du processus d'indexation

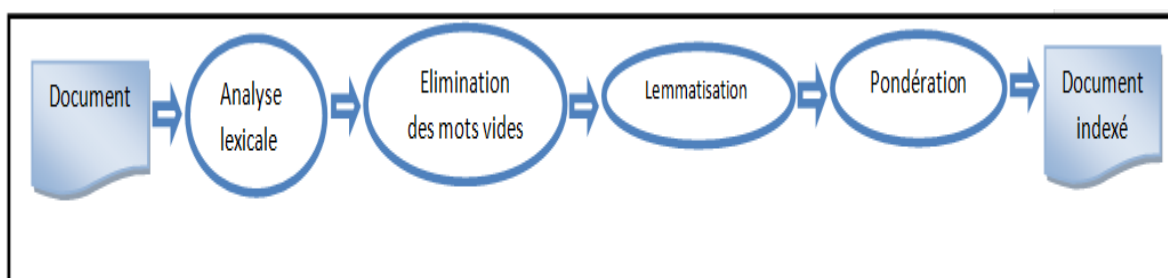


Figure 1.3 : les étapes du processus d'indexation

- 1) **L'Analyse lexicale** : elle permet de reconnaître les espaces de séparation, des chiffres, des mots et des ponctuations.
- 2) **Elimination des mots vides** : elle permet d'améliorer la fiabilité d'un SRI au sens de qualité logiciel (temps d'exécution) et de performances.
- 3) **Lemmatisation (Radicalisation)**: consiste à éliminer les terminaisons des mots, et garder seulement leur racine (lemme) pour avoir une forme identique de ces mots.

Plusieurs méthodes sont utilisées pour retrouver la racine lexicale d'un mot tel que : l'algorithme de porter, la troncature et la méthode des n-grammes. (SHANNON, 1948).

4) **La Pondération** : Elle permet de retrouver les documents qui répondent exactement au besoin exprimé par la requête mais aussi les documents qui répondent partiellement ou approximativement à ce besoin. [7]

a) **Les types de pondération** : On distingue deux types de pondération:

➤ **La pondération Locale (TF)**: issue des travaux de Luhn (1958) qu'on peut la calculer comme la fréquence du terme dans le document (**Terme Frequency TF**). Il s'agit tout simplement du nombre d'occurrences du terme dans le document.

➤ **La pondération Globale (IDF)**: est souvent attribuée à Spärck Jones (1972) appelée aussi la fréquence inverse du document (**Inverse document frequency IDF**), il s'agit d'une mesure qui permet d'évaluer l'importance du terme par rapport au corpus. Elle consiste à calculer le logarithme du nombre total de documents sur le nombre de documents contenant le terme :

$$idf_i = \log \frac{|D|}{|\{d_j: t_i \in d_j\}|} \quad (1.3)$$

Où :

$|D|$: nombre total de documents dans le corpus ;

$|\{d_j: t_i \in d_j\}|$: Nombre de documents où le terme t_i apparaît.

➤ **TF*IDF**: ce poids est tout simplement le produit des deux quantités TF et IDF :

$$tf * idf_{i,j} = tf_{i,j} * idf_i \quad (1.4) [7]$$

Les pondérations citées ci-dessus sont les plus utilisées dans la littérature. Néanmoins, Il existe d'autres pondérations telles que TFC et LTC [Aas, Eikvil 1999]. où la première est semblable au TF*IDF à la différence que la TFC emploie la longueur normalisée des documents, ainsi que la deuxième emploie le logarithme de la fréquence d'une unité lexicale. [6]

$$\bullet \quad TFC_{ij} = \frac{tf_{ij} * idf_i}{\sqrt{\sum_{k=1}^M (tf_{kj} * idf_k)^2}} \quad (1.5)$$

$$LTC = \frac{\log(tf_{ij}+1)*idf_i}{\sqrt{\sum_{k=1}^M (\log(tf_{kj}+1)*idf_k)^2}}$$

(1.6)

- **BM25** : Okapi BM25 est une méthode de pondération (fonction de classement) utilisée en recherche d'information. Elle est une application du modèle probabiliste de pertinence, proposé en 1976 par Robertson et Jones. qui classe un ensemble de documents en fonction des termes de requête apparaissant dans chaque document, indépendamment de l'interdépendance entre les termes de requête dans un document (par exemple : leur proximité relative). Il est utilisé par les moteurs de recherche pour classer les documents correspondants en fonction de leur pertinence par rapport à une requête de recherche donnée et est souvent appelé «Okapi BM25», car le système de recherche d'informations Okapi a été le premier système implémentant cette fonction. La formule d'extraction BM25 appartient à la famille BM des modèles d'extraction (BM signifie Best Match). Ce n'est pas une fonction unique, Il existe toute une famille de fonctions attribuant un score à chaque document pour une requête donnée. [w3]

1.6.2. L'appariement document-requête

Le processus d'appariement document-requête permet de mesurer la pertinence d'un document vis-à-vis d'une requête. De manière générale, à chaque réception d'une requête, le SRI calcule un score de pertinence. Ce score de pertinence est calculé à partir d'une fonction ou d'une mesure de similitude, notée $RSV(Q;D)$ où Q est une requête, D un document de la collection et RSV (*Retrieval Status Value*) est une fonction de correspondance qui consiste à comparer les représentations des documents à celle de la requête afin de retrouver des documents recherchés . Il existe deux méthodes d'appariement :

- **Appariement exact**: le résultat est une liste de documents respectant exactement la requête spécifiée avec des critères précis.
- **Appariement approché**: le résultat est une liste de documents sensés être pertinents pour la requête, il peut inclure une estimation de l'appariement. Les documents peuvent être ordonnés afin de refléter la pertinence des documents vis-à-vis de la requête. [10]

1.6.3. La reformulation de requête

La reformulation de requête est proposée comme une méthode pour la recherche d'information adaptative. C'est un processus permettant de générer une requête plus adéquate à la recherche d'information dans l'environnement du SRI que celle initialement formulée par l'utilisateur. Son principe est de modifier la requête de l'utilisateur par ajout de termes significatifs et/ou ré-estimation de leur poids.

Nous distinguons principalement deux types de reformulation de requête :

- **La reformulation directe** : qui est un processus de repondération de la requête initiale en utilisant des critères de choix définis sans l'intervention de l'utilisateur.
- **La reformulation par injection de pertinence** : est une forme de recherche évolutive et interactive; elle procède à la modification de la requête initiale en termes et poids, sur la base des jugements de pertinence de l'utilisateur sur les documents restitués par le SRI. Son principe fondamental est d'utiliser la requête initiale pour amorcer la recherche d'information, puis exploiter itérativement les jugements de pertinence de l'utilisateur afin d'ajuster la requête par expansion, repondération, ou combinaison des deux procédures, en direction des documents pertinents. [10]

❖ Les approches pour la reformulation de requêtes :

➤ La reformulation manuelle

Cette approche est associée aux systèmes de recherche booléens. On peut procéder à la reformulation de requête en utilisant un vocabulaire contrôlé (thésaurus ou classification) pour permettre à l'utilisateur de trouver les bons termes pour compléter sa requête.

➤ La reformulation automatique

Lorsque le feedback de pertinence s'accompagne d'une adjonction (et/ou) suppression de termes, on parle de reformulation automatique. La requête de l'utilisateur est remaniée automatiquement, pour intégrer les descripteurs des documents jugés pertinents ou rejetés. On trouve différentes variantes de cette technique : celles qui sont utilisées automatiquement pour reformuler la requête en augmentant le poids des termes présents dans les documents jugés pertinents et inversement pour diminuer les poids des termes jugés non pertinents.

➤ La reformulation interactive

Dans une reformulation interactive, l'utilisateur joue un rôle crucial. A l'inverse de la reformulation automatique, dans ce type de reformulation l'utilisateur et le système sont, les deux, responsables de la détermination et du choix des termes candidats à la reformulation. Le système joue un grand rôle dans la suggestion des termes, le calcul des poids des termes et l'affichage à l'écran de la liste ordonnée des termes. L'utilisateur examine cette liste et décide du choix des termes à ajouter dans la requête. Donc, c'est l'utilisateur qui prend la décision finale dans la sélection des termes. [11]

1.7. Les systèmes de recherche d'information

1.7.1. Définition

Un Systèmes de recherche d'information (SRI) est un ensemble de techniques qui assurent les fonctions nécessaires pour la RI. Il a pour rôle de sélectionner les documents qui peuvent répondre au besoin en information de l'utilisateur formulé par une requête de recherche. Cette requête est souvent sous forme d'un langage naturel, c'est-à-dire un ensemble de mots clés, la requête est considérée dans ce cas comme basique, ou peut être sous une composition de requêtes basiques avec un ensemble d'opérateurs logiques (ET, OU, SAUF, etc.) appelée *requête logique*. [3]

La (figure 4.1) il représente l'objectif des SRI conçus pour aider les utilisateurs à trouver les documents qu'ils souhaitent et à mieux répondre à la requête d'un utilisateur.

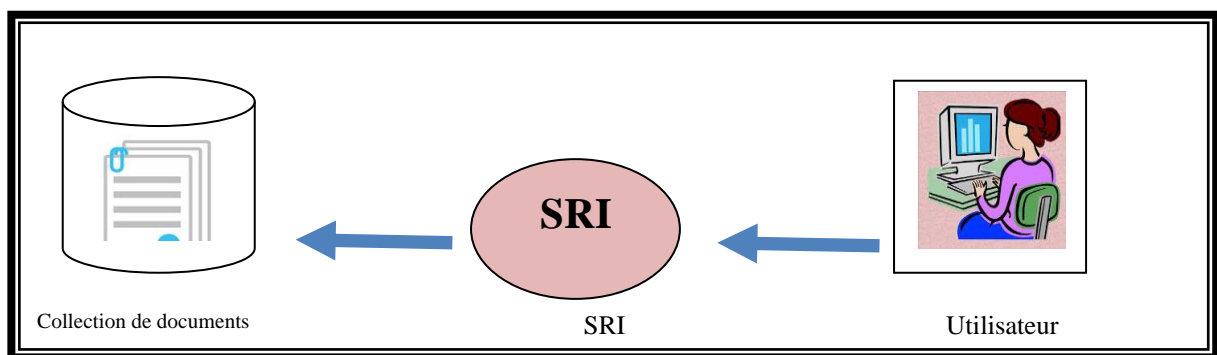


Figure 4.1 : définition d'un système de recherche d'information [w4]

Un SRI a pour fonction de permettre à l'utilisateur d'accéder à des documents qui contribuent à combler son besoin en information, exprimé sous forme de requête, qui motive sa recherche. Ainsi, le système peut être vu par l'utilisateur comme un instrument

de prédiction de la pertinence des documents d'un corpus par rapport à sa requête. Pour évaluer l'adéquation entre un ensemble de documents et une requête, le SRI doit posséder d'une part une représentation interne des documents disponibles et de la requête utilisateur et d'autre part d'une méthode de comparaison afin de déterminer leur degré de correspondance. Les représentations internes ainsi que la manière de les comparer définissent le modèle de recherche. [1]

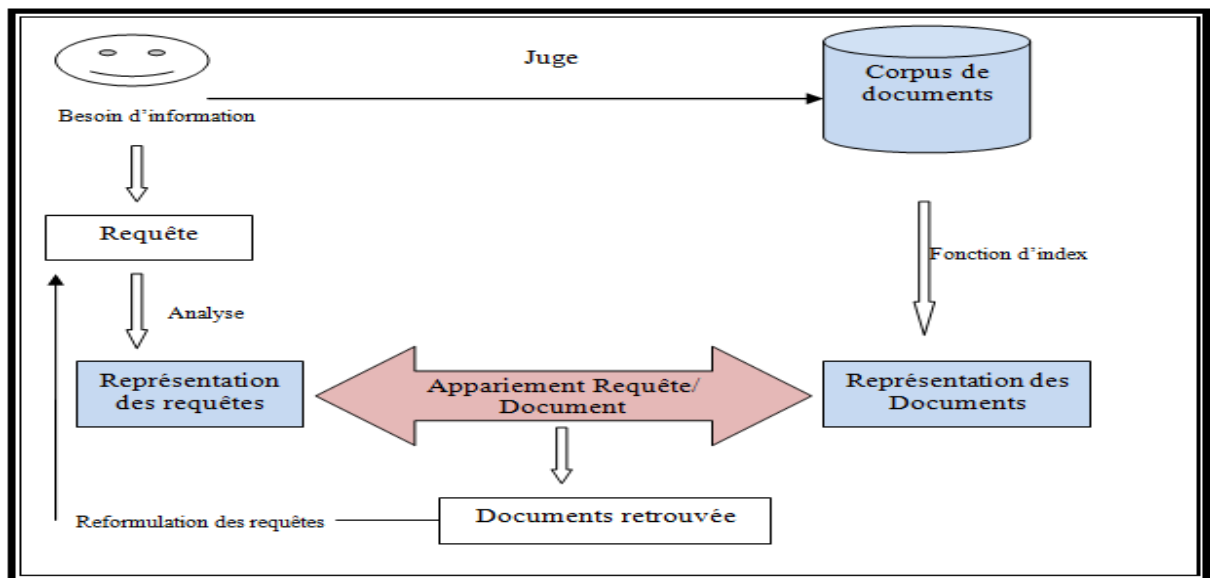


Figure 5.1 : Processus de RI et le système de recherche d'information [w5]

1.7.2. Modélisation du système de recherche d'information

Un modèle théorique doit donner une interprétation précise du poids d'un terme de l'index, il doit aussi prendre en compte, à défaut de générer les relations possibles entre les termes d'indexation.

Un modèle détermine la ressemblance entre un document et une requête à partir de leurs représentations respectives, ce qui permet d'établir un lien entre la représentation d'une requête et la représentation d'un document. Donc, il est utilisé pour comparer les index et ainsi estimer la similarité entre requête et documents.

Il existe 3 principaux modèles dans la RI: Le modèle booléen, le modèle vectoriel et le modèle probabiliste. Le Figure suivante, présente les modèles les plus utilisés dans la littérature par les SRI. Pour plus de détails sur les différents modèles veuillez consulter [7]

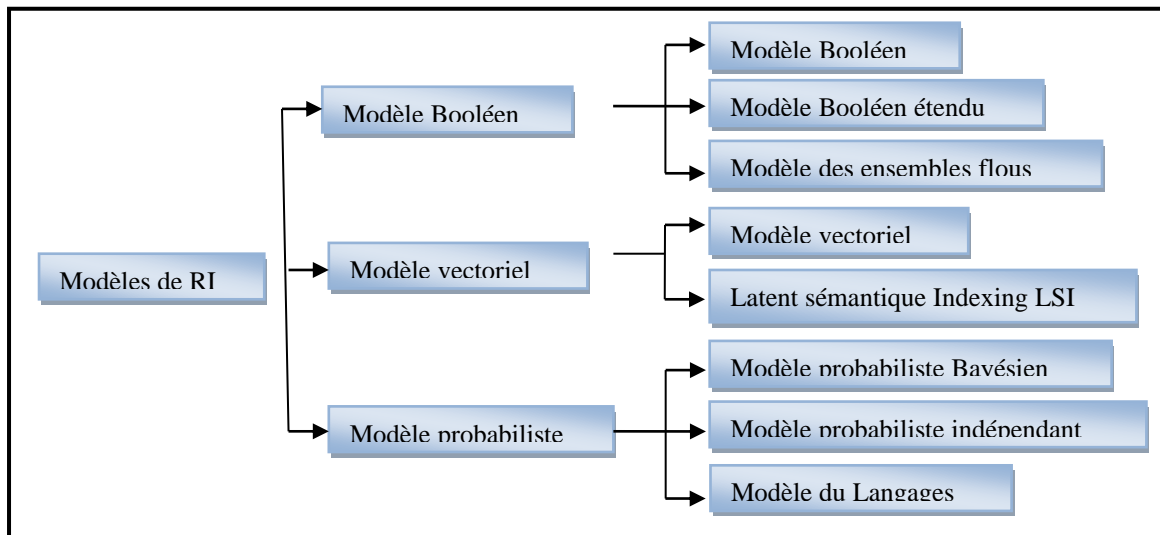


Figure 1.6 : Taxonomie des modèles de RI proposés [3]

❖ **Modèle booléen (Booléen Model MB booléen (±1950))**

Dans l'histoire de la RI, le modèle booléen proposé par Salton et McGill (1950) fut le premier modèle proposé. L'adjectif « booléen » fait référence à l'usage de l'algèbre de Boole. Un SRI booléen retourne les documents qui contiennent un ou plusieurs termes de la requête. Ainsi un document est soit pertinent soit non pertinent par rapport à une requête donnée. Pour pallier cette absence de nuances dans la pertinence d'un document par rapport à une requête, une extension de ce modèle nommée « booléen étendu (*extended boolean model*) » a été proposée permettant de prendre en compte la pondération des termes.

Il s'agit du modèle le plus simple. Il repose sur les opérateurs de l'algèbre de Boole (and, or, xor ...). C'est un modèle simple à implémenter, il est très efficace pour des corpus examinés par des experts. Cependant, il présente deux inconvénients majeurs :

- Il ne peut pas répondre partiellement à un besoin exprimé par une requête.
- Il ne permet pas de classer les documents par rapport à leurs pertinences vis-à-vis d'une requête.

Ce modèle permet de répondre à des requêtes du genre « manger et pomme » « pomme ou banane ». [7]

❖ *Modèle Probabiliste (probabilistic models 1976)*

Le modèle Probabiliste proposée en 1976 par Robertson et Jones. Apparu consécutivement au modèle vectoriel, permet de quantifier l'incertitude dans la représentation des informations ainsi que l'imprécision dans l'expression des besoins. Les différents modèles peuvent être classés en fonction des théories mathématiques sur lesquelles ils se basent.

Le modèle probabiliste est utilisé pour estimer la probabilité qu'un document d soit retourné par le système sachant une requête q . Il s'agit d'estimer la probabilité pour un document qu'il soit pertinent par rapport à une requête.

La similarité entre D et Q est calculée comme suit :

$$Sim(Q, D) = \frac{P(R|D)}{P(NR|D)} = \frac{P(D|R)*P(R)}{P(D|NR)*P(NR)} \quad (1.8)$$

Notons :

R = pertinence

NR = non pertinence

$P(R|D)$ = probabilité que d soit pertinent

Le modèle probabiliste est fortement dépendant des probabilités initiales et considère que les termes sont tous indépendants. [7]

❖ *Modèle Vectoriel (Vector Space Model) (VSM)*

Le modèle vectoriel (Proposé par Gerard Salton, 1970) : est le premier à intégrer un élément fondamental : la capacité d'ordonner les documents restitués selon un critère de pertinence. Ce modèle est considéré le meilleur par rapport aux deux précédents. Il s'agit d'une représentation mathématique des contenus des documents et des requêtes selon une approche algébrique.

Chaque document est composé d'un ensemble de termes descripteurs. Chaque descripteur est présent une ou n fois. Les descripteurs d'un document sont représentés dans un repère dont la dimension égale à la taille du vocabulaire et la longueur de chaque vecteur dépend de la fréquence du descripteur correspondant. Les documents et la requête sont donc représentés dans ce même repère. La similarité entre un document et une requête est estimée par l'angle qui sépare leurs vecteurs :

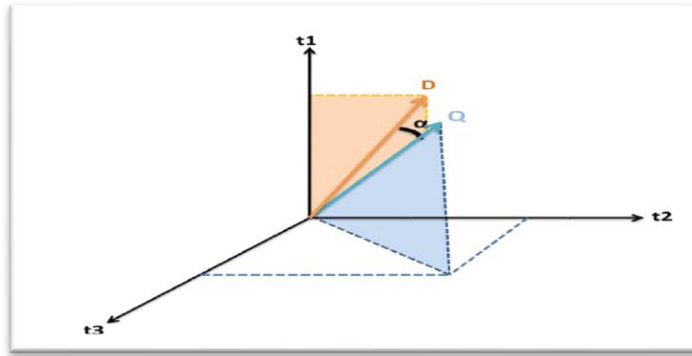


Figure 1.7 : Similarité selon le modèle vectoriel

La proximité d'une requête Q à un document D sera ainsi donnée par:

$$\cos \alpha = \frac{D \cdot Q}{\|D\| \|Q\|} \quad (1.9)$$

Plus ce cosinus se rapproche de 1, plus le document est jugé pertinent par rapport à la requête. [7]

1.7.3. Evaluation des systèmes de recherche d'information

Le but de la RI est de trouver des documents pertinents à une requête, et donc utiles pour l'utilisateur. La qualité d'un système doit être mesurée en comparant les réponses du système avec les réponses idéales que l'utilisateur espère recevoir. Plus les réponses du système correspondent à celles que l'utilisateur espère, mieux est le système. [5]

La validation d'un SRI se base sur l'évaluation expérimentale de ses performances. L'évaluation peut couvrir plusieurs critères d'efficience et d'efficacité qui sont en général construits à partir des jugements exprimés par des utilisateurs ou par des experts. On peut citer : la pertinence des résultats, la qualité de la présentation des résultats, la performance qui touche à son tour plusieurs critères concernant la consommation de ressources, telle que le temps de réponse, l'espace mémoire, la capacité en charge, etc. Le critère le plus important qui sans doute intéresse le plus l'utilisateur est celui qui mesure la capacité qu'un SRI puisse satisfaire son besoin en information. Il s'agit de la pertinence des résultats renvoyés par ce système. La liste des résultats peut être divisée en quatre ensembles: pertinents ou non (l'utilisateur dit Oui / Non), sélectionnés ou non (le système dit Oui / Non). Ces différentes situations sont résumées dans le tableau suivant, appelé la table de contingence.

	Pertinent	Non pertinent
Sélectionné	Vrai positif	Faux positif
Non sélectionné	Faux positif	Vrai négatif

Tableau 1.1: La table de contingence

À partir de ces valeurs, différentes métriques peuvent être calculées, permettant d'évaluer la pertinence des résultats, nous citons les deux métriques les plus utilisées en RI, à savoir, les taux de précision et de rappel.

- **La précision (Precision):** tente de trouver combien d'éléments sont pertinents parmi ceux qui sont sélectionnés, elle mesure la capacité du système à rejeter tous les documents non pertinents.

$$\text{Précision } (P) = \frac{\text{vrai positifs}}{\text{vrai positifs} + \text{faux positifs}} = \frac{\text{nombre de documents pertinents sélectionnés}}{\text{Nombre de documents sélectionnés}}$$

(1.10)

- **Le rappel (Recall):** tente de trouver combien d'éléments pertinents sont sélectionnés, il mesure la capacité du système à sélectionner tous les documents pertinents.

$$\text{Rappel } (P) = \frac{\text{vrai positifs}}{\text{vrai positifs} + \text{faux négatifs}} = \frac{\text{nombre de documents pertinents sélectionnés}}{\text{Nombre de tous les documents pertinents}}$$

(1.11) [3]

- **La mesure F1:** [van Rijsbergen, 1979] Mesure de la performance d'un système ou d'un algorithme qui correspond à la moyenne harmonique du taux de précision et du rappel. Pour utiliser cette mesure, il est donc nécessaire de fixer préalablement un seuil de décision pour le classement, et de calculer la valeur de F pour ce seuil.

Les valeurs obtenues servent à évaluer la performance globale d'un système ou d'un algorithme et varient de 0 (pire performance) à 1 (performance parfaite).

Ces deux mesures varient en sens inverse : les méthodes permettant d'augmenter la précision ont tendance à dégrader le rappel et vice versa. La mesure F1 permet de combiner le rappel et la précision comme suit :

$$F1 = 2 \frac{\text{Rappel } (P) \cdot \text{Précision } (P)}{\text{Rappel } (P) + \text{Précision } (P)} \quad (1.12) [12]$$

1.8. Domaines d'application de la RI

La RI est un domaine très large, et qui peut être utilisée dans plusieurs type d'applications afin de répondre aux besoins en information des utilisateurs. Citons comme exemples : Les systèmes de recherche Ad Hoc pour la recherche des documents pertinents dans des collections fixes, les systèmes « Question\Réponse » pour l'extraction des réponses à partir de documents récupérés, les annuaires, la classification de documents, la catégorisation de documents, la synthèse de documents pour l'extraction des informations à partir des documents restitués, et la recherche dans les bases de données structurées. [6]

Partie 2: l'apprentissage par renforcement profond

Les algorithmes d'apprentissage par renforcement (RL) impliquent la stratégie d'apprentissage via l'interaction (séquences d'actions, observations et récompenses) avec l'environnement. Les méthodes basées sur l'apprentissage par renforcement ont connu de grands succès dans une variété de tâches allant de la robotique à l'allocation des ressources. Celles-ci en ont fait l'un des principaux candidats prometteurs pour atteindre l'objectif de l'intelligence artificielle (IA), la construction de ces agents autonomes qui peuvent apprendre dans des environnements incertains. Avant qu'un agent ou un robot (logiciel ou matériel) puisse sélectionner une action, il doit avoir une bonne représentation de l'environnement dans lequel l'agent doit être appris. Ainsi, la perception est l'un des problèmes clés qui doivent être résolus avant que l'agent puisse décider de sélectionner une action optimale à prendre.

La représentation de l'environnement peut être donnée ou acquise. En renfort les tâches d'apprentissage, généralement un expert humain fournit des caractéristiques artisanales de l'environnement sur la base de sa connaissance de la tâche. De plus, la performance de l'apprentissage de cette manière est intensément en fonction de la qualité de la représentation de la fonction. Plus loin, la généralisation d'un une méthode donnée pour différentes tâches peut ne pas fonctionner correctement. Par conséquent, cela devrait être fait automatiquement, puisque la précision, que l'extraction automatique des fonctionnalités peut fournir, est bien meilleure, et aussi l'algorithme ne souffrira pas des problèmes de fonctionnalités conçues à la main.

Ces dernières années, les recherches effectuées dans le domaine de l'apprentissage en profondeur ont montré qu'il est un outil puissant pour effectuer une extraction automatique des caractéristiques à partir de données brutes, par exemple, des pixels bruts d'une image.

Il a gagné une énorme attraction non seulement dans les communautés universitaires (en raison de sa performance dans de nombreuses applications telles que la reconnaissance de formes, la reconnaissance vocale, la vision par ordinateur et le traitement du langage naturel), mais il a été appliqué également avec succès dans les grands produits de la technologie comme Google (traducteur Google, moteur de recherche d'images), Apple (Siri d'Apple), Microsoft (recherche vocale Bing) et d'autres grandes sociétés telles que Facebook et IBM.

Récemment, certaines techniques d'apprentissage en profondeur, notamment les architectures profondes supervisées et non supervisées, y compris les perceptrons multicouches (MLP), les réseaux de neurones convolutionnels (CNN), les auto-encodeurs et les réseaux de neurones récurrents (RNN) ont commencé à intégrer les méthodes d'apprentissage par renforcement. L'évaluation des nouveaux algorithmes et méthodes résultants a montré que les techniques d'apprentissage en profondeur peuvent également être utilisées pour apprendre des représentations utiles pour les problèmes d'apprentissage par renforcement. Combiner RL et deep learning permet à un agent RL d'avoir une bonne perception de son environnement en utilisant les possibilités que les réseaux de neurones profonds peuvent offrir. [13]

1.9. Concepts connexes:

1.9.1. L'intelligence artificielle (IA) : L'intelligence artificielle est la science dont le but est d'utiliser une machine pour faire des tâches que l'homme accomplit en utilisant son intelligence. La terminologie de l'Intelligence Artificielle est apparue en 1956, et elle est nommée aussi *l'Informatique Heuristique*. [14]

1.9.2. L'apprentissage automatique (Machine Learning): La définition de l'apprentissage automatique «*machine learning*» (Mitchell «Machine Learning» 1997) est une « imposture » scientifique (*how to make computers learn*) [15]

Est utilisé pour concevoir des modèles complexes et des algorithmes qui conduisent eux-mêmes à des prédictions. [16]

1.9.3. L'apprentissage en profondeur

❖ Définition :

L'apprentissage profond est un ensemble de techniques de l'apprentissage automatique qui a permis des avancées importantes en intelligence artificielle dans les dernières années.

Dans l'apprentissage automatique, un programme analyse un ensemble de données afin de tirer des règles qui permettront de tirer des conclusions sur de nouvelles données.

L'apprentissage profond est basé sur ce qui a été appelé, par analogie, des « réseaux de neurones artificiels », composés de milliers d'unités (les « neurones ») qui effectuent chacune de petites opérations simples. Les résultats d'une première couche de « neurones » servent d'entrée aux calculs d'une deuxième couche et ainsi de suite. Les progrès de l'apprentissage profond ont été possibles notamment grâce à l'augmentation de la puissance des ordinateurs et au développement de grandes bases de données « big data ». [17]

1.9.4. L'Apprentissage par renforcement :

❖ *Définition :*

L'apprentissage par renforcement (Sutton & Barto, 1998) : est un domaine de l'apprentissage automatique qui se concentre sur l'utilisation des récompenses afin de former les agents à agir dans des environnements. L'apprentissage par renforcement fournit une structure où l'agent se trouve dans un environnement dans lequel il peut prendre des mesures, puis faire des observations et recevoir des récompenses. Ces récompenses peuvent ensuite être utilisées pour améliorer la politique de l'agent pour choisir des actions à l'avenir, dans le but de maximiser les récompenses qu'il reçoive. [18]

Il est considéré aussi comme une branche de l'apprentissage automatique concernée par la prise de séquences d'actions.

Généralement, il est décrit comme un agent interagissant avec un environnement auparavant inconnu, en essayant de maximiser des récompenses cumulatives.

En d'autres termes, l'agent, en observant les résultats des actions qu'il entreprend dans l'environnement, essaie d'apprendre une séquence optimale d'actions à exécuter afin d'atteindre son objectif. [13]

❖ *Modélisation avec l'apprentissage par renforcement :*

Un agent d'apprentissage par renforcement peut être modélisé comme un processus de décision de Markov (MDP). Si les états et les espaces d'action sont nis, alors le problème est appelé un MDP fini. [13]

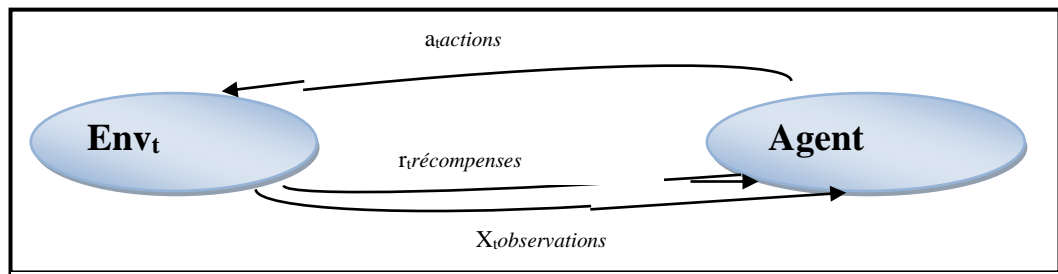


Figure 1.8 : Modélisation d'un agent d'apprentissage par renforcement [w6]

- *Observations:* images de caméra, angles articulaires
- *Actions:* couples conjoints
- *Interactions entre l'agent et l'environnement dans un cadre d'apprentissage par renforcement au temps t :* L'agent reçoit une observation x_t de l'environnement, qui lui permet de choisir une action a_t et de l'effectuer dans l'environnement. Il reçoit ensuite une récompense r_{t+1} .

❖ **Exemples d'utilisation de l'apprentissage par Renforcement :**

- Contrôler les systèmes physiques: marcher, conduire, nager, ...
- Interagir avec les utilisateurs: fidéliser les clients, personnaliser le canal, optimiser l'expérience utilisateur, ...
- Résoudre des problèmes logistiques: planification, allocation de bande passante, contrôle d'ascenseur, radio cognitive, optimisation de puissance, etc...
- Jouer à des jeux: échecs, dames, Go, jeux Atari, ...
- Apprenez les algorithmes séquentiels: attention, mémoire, conditionnel calcul, activations,... [20]

1.9.5. L'apprentissage par renforcement profond

❖ **Définition:**

L'apprentissage par renforcement profond (*Deep Reinforcement Learning DRL*) est la combinaison de l'apprentissage par renforcement (RL) et de l'apprentissage en profondeur. [21]

On peut dire aussi que c'est l'apprentissage par renforcement appliqué à l'aide de réseaux de neurones profonds. Ce type d'apprentissage implique que les ordinateurs agissent sur des modèles sophistiqués et examinent un grand nombre d'entrées afin de déterminer un chemin ou une action optimisée. [22]

Utiliser de telles structures permet de représenter des fonctions sur un espace d'entrée à dimensionnalité élevée en extrayant automatiquement une hiérarchie de descripteurs. De plus, les dimensions des espaces d'états des tâches d'apprentissage par renforcement sont de plus en plus élevées. [22]

❖ *Les algorithmes d'apprentissage par renforcement profond*

Le DRL fait appel aux algorithmes d'apprentissage par renforcement, dont les deux les plus utilisés sont le TD learning (pour temporal différence learning) et le Q-learning. Des modèles d'apprentissage qui s'inspirent du processus humain d'acquisition des connaissances par essais-erreurs, et qui valident les décisions prises par la machine via un mécanisme de récompense pour orienter les choix dans la direction voulue.

Le Réseau Q profond (DQN) est l'un des premiers algorithmes RL profonds réussis où les réseaux de neurone convolutionnels (CNN +REF) sont utilisés comme approximateur de fonction pour l'algorithme RL classique. [23]

1) **L'algorithme DQN**

La première application d'apprentissage par renforcement profond appris sur des pixels bruts et sans calculer à la main des descripteurs. Cette approche utilise un réseau de neurones convolutionnel pour approximer la fonction Q.

L'algorithme DQN combine le Q-learning avec CNN. En Q-learning, l'agent apprend une fonction de valeur optimale p2405

consiste à trouver une politique qui donne un maximum de récompenses. Les fonctions valeurs $V\pi$ définissent un ordre partiel sur les politiques π . Une politique π est supérieure à une politique π' si $V\pi'(s) \leq V\pi(s)$ pour les états possibles. On note cela $\pi' \leq \pi$. Il existe au moins une politique qui est meilleure que toutes les autres, appelées politiques optimales et notée π^* . Ces politiques partagent la même fonction valeur d'état, notée V^* , appelée fonction de valeur d'état optimale. On a :

$$V^*(s) = \max_{\pi} V^{\pi}(s) \quad (1.13)$$

Les politiques optimales partagent aussi une même fonction de valeur d'action optimale, notée Q^* et définie par :

$$Q^*(s, a) = \max_{\pi} Q^{\pi}(s, a) \quad (1.14)$$

On a l'équation suivante :

$$Q^*(s, a) = E \{r_{t+1} + \gamma V^*(s_{t+1}) | s = s_t, a = a_t\} \quad (1.15) [24].$$

Dans DQN, un CNN à trois couches suivies de deux couches entièrement connectées (paramétré comme θ) sont utilisées pour approximer $Q(s, a; \theta) \approx Q^*(s, a)$ puisque l'espace d'état des entrées brutes est trop grande pour calculer directement les valeurs Q . [23]

- 2) **Le Modèle BERT** : BERT est un Framework d'apprentissage automatique open source pour le traitement du langage naturel (NLP).est conçu pour aider les ordinateurs à comprendre la signification d'un langage ambigu dans le texte en utilisant le texte pour établir le contexte. Le cadre BERT a été pré-formé à l'aide de texte de Wikipédia et peut être affiné avec des ensembles de données de questions et réponses.

BERT il est conçu pour lire dans les deux sens à la fois, est connue sous le nom de bidirectionnalité, est un modèle d'apprentissage en profondeur dans lequel chaque élément de sortie est connecté à chaque élément d'entrée, et les pondérations entre eux sont calculées dynamiquement en fonction de leur connexion. (En PNL, ce processus est appelé l'attention), introduits pour la première fois par Google en 2017, les modèles de langage utilisaient principalement des réseaux de neurones récurrents (RNN) et des réseaux de neurones convolutifs (CNN) pour gérer les tâches PNL.

- 3) **Les Réseaux de Neurone Convolutionnel (CNN)** :

Un réseau de neurones convolutifs (CNN) est l'un des algorithmes les plus populaires pour l'apprentissage en profondeur, dans lequel le modèle apprend à effectuer des tâches de classification directement à partir d'images et de vidéos, de textes ou de sons.

Par exemple, les applications d'apprentissage en profondeur utilisent les CNN pour analyser des milliers de rapports de pathologie afin de détecter visuellement les cellules cancéreuses.

Un réseau de neurones convolutifs peut contenir des dizaines, voire plusieurs centaines de couches.

Les CNN identifient et catégorisent les images, le texte, l'audio et la vidéo.

Comme d'autres réseaux de neurones, les CNN se composent d'une couche d'entrée, d'une couche de sortie et de nombreuses couches cachées entre ces deux couches. [w7]

4) **Les Réseaux de Neurone Récurent (RNN) :**

Les réseaux neuronaux récurrents: l'un des algorithmes derrière réalisations de Deep Learning ces dernières années. en raison d'une mémoire interne, Les réseaux neuronaux récurrents produisent des résultats prédictifs dans des données séquentielles que d'autres algorithmes ne peuvent pas produire.

C'est la raison pour laquelle ils sont l'algorithme préféré pour les données séquentielles comme les séries temporelles, la parole, le texte, les données financières, audio, vidéo, météo.

Les données séquentielles est essentiellement de données ordonnées, où les choses liées se suivent. Le type le plus courant de données séquentielles est peut-être les données de séries chronologiques. [w8]

❖ *Quelques modèles d'apprentissage par renforcement profond :*

- *TD-Gammon :*

L'algorithme d'apprentissage par renforcement le plus connu qui utilise des réseaux de neurones (avec une seule couche cachée) est le joueur de backgammon RL de classe mondiale nommé TD-Gammon. TD-Gammon utilise l'algorithme TD (λ) pour entraîner un réseau neuronal profond à apprendre à jouer au backgammon. [13]

❖ *Le calculateur AlphaGo*

DeepMind est un système populaire qui utilise cette méthode. Le géant américain s'est basé sur la plateforme d'IA de Google pour développer AlphaGo, le supercalculateur connu pour avoir battu en 2017 le champion du monde de go, le Chinois Ke Jie.

❖ *La start-up américaine Prowler.io*

Aux côtés de la robotique industrielle, l'apprentissage par renforcement profond monte en puissance sur d'autres créneaux. Par exemple dans le monde financier, pour optimiser les ventes automatisées ou la gestion des risques de marché la start-up américaine Prowler.io a été inventée.

Le DRL est utilisé aussi dans les chatbots et autres systèmes conversationnels intelligents.

Les géants du numérique Apple et Microsoft par exemple y voient un levier pour booster leurs assistants vocaux. Le premier a acquis en 2016 la technologie de DRL britannique VocallIQ qu'il a depuis intégrée à Siri. Quant à Microsoft, il a racheté fin 2017 le Canadien Maluuba, autre spécialiste du domaine. Ce dernier a rejoint sa R&D centrée sur l'IA. [w9]

❖ **Une approche d'apprentissage par renforcement profond**

Récemment, il y a quelques exemples notables: politiques visuo-motrices profondes.

DRL a été étudié dans les tâches de PNL. Par exemple, récemment, DRL a été étudié pour un problème d'extraction d'informations. Ils ont conçu un cadre qui peut décider d'acquiescer des preuves externes et le cadre est sous le renforcement méthode d'apprentissage.

De récents travaux d'apprentissage en profondeur ont également examiné transfert d'apprentissage. Plus récent le travail en apprentissage profond a également envisagé le transfert politique en réutilisant les paramètres de politique entre environnements, en utilisant soit la régularisation soit le roman architecture de réseaux de neurones, bien que ce travail n'ait pas examiné les stratégies d'apprentissage actif de transfert entre langues avec espace de fonctionnalité partagé en état. [25]

Partie 3: *La recherche d'information et l'apprentissage par renforcement profond*

Récemment, et en vue sa grande performance, l'apprentissage par renforcement profond est devenu un axe de recherche dans plusieurs domaines, y compris le domaine de la recherche d'information. Dans cette partie, nous citons quelques travaux de la recherche d'information qui ont utilisé cette méthode.

1.10. Travaux connexes

1.10.1. Un agent d'apprentissage par renforcement pour le filtrage personnalisé des informations : [Seo and Zhang, 00]

Une méthode d'apprentissage des intérêts des utilisateurs dans un système de filtrage d'informations personnalisé basé sur le Web appelé WAIR (Web-Agents for Information Retrieval). a été proposée par [Seo and Zhang, Actes de la 5e conférence internationale du 2000]. La méthode analyse les réactions des utilisateurs aux documents présentés et apprend les profils pour les utilisateurs individuels. L'apprentissage par renforcement est utilisé pour adapter la pondération des termes dans le profil utilisateur afin que les préférences de l'utilisateur soient mieux représentées à partir des observations directes des comportements de l'utilisateur pendant l'interaction.

Dans ce modèle, Rocchio a suggéré une méthode de retour d'information sur la pertinence comme suit:

$$Q' = Q + \frac{1}{n_1} \alpha R_i - \frac{1}{n_2} \beta \sum_{i=1}^{n_2} (S_i) \quad (1.16)$$

Où Q est le vecteur de la requête initiale,

R_i est le vecteur du document i pertinent,

S_i est le vecteur du document i non pertinent, et α, β sont les poids de Rocchio. Q' est le vecteur modifié de la requête d'origine plus les vecteurs des documents pertinents et non pertinents.

Ide a défini trois stratégies particulières prolongeant le travail de Rocchio. Pour comparer ses performances dans l'expérience, nous décrivons l'une des méthodes ici:

$$Q' = \alpha Q + \beta \sum_{i=1}^{n_1} (R_i - S_i) \quad (1.17)$$

où S_i est le document non pertinent le mieux classé.

Le système proposé exploite une méthode qui reflète l'opinion directement aux termes du profil. Autrement dit, si un terme dans le document estimé comme «pertinent» est inclus dans le profil de l'utilisateur, le poids du terme est mis à jour comme suit:

$$w_{pk} \leftarrow w_{pk} + \beta r_i, \text{ Si } k \in D_i \quad (1.18)$$

Afin d'évaluer ce système, 7 personnes se sont portées volontaires pour suggérer 14 sujets, où les 14 rubriques représentent un total de 5 600 documents HTML. Les expériences ont montré que la méthode peut fournir des documents qui sont plus pertinents pour les intérêts spécifiques de l'utilisateur par rapport à d'autres méthodes de rétroaction. En plus, elle s'adapte efficacement aux intérêts spécifiques de rétroactions implicites uniquement. [26]

1.10.2. Optimisation des évaluations de requêtes à l'aide de l'apprentissage par renforcement pour la recherche sur le Web [Corby Rosset et al., 18]

Un système de recherche d'information, *Bing*, qui utilise l'apprentissage par renforcement dans l'étape d'appariement a été proposé par [Rosset et al., 2018]. *Bing* maintient un index inversé par champ, et la liste de publications correspondante peut être organisée pour chaque terme en fonction de mesures au niveau du document, et la requête est classée dans l'une des rares catégories prédéfinies.

Dans *Bing*, la génération de candidats implique l'analyse de l'index à l'aide de plans de correspondance conçus statiquement qui prescrivent des séquences de différents critères de correspondance et de conditions d'arrêt. Les documents sont analysés, aussi, en fonction du plan de correspondance choisi qui consiste en une séquence de règles de correspondance et des critères d'arrêt correspondants.

La planification des correspondances est définie comme une tâche d'apprentissage par renforcement (RL), pour apprendre une politique qui décide séquentiellement quelles règles de correspondance utiliser lors de la génération des candidats. Le modèle est formé pour maximiser une récompense cumulative calculée en fonction de la pertinence estimée des documents supplémentaires découverts, actualisés par leur coût de récupération.

Bing utilise un cadre télescopique pour élaguer de manière itérative l'ensemble des documents candidats considérés pour une requête. Lors de la réception d'une demande de recherche, la requête classe en fonction d'un ensemble de caractéristiques disponibles sachant : la popularité de la requête, le nombre de termes de la requête et la fréquence des termes de la requête dans le document. [27]

1.10.3. Apprentissage par renforcement profond pour la recherche, la recommandation et la publicité en ligne: une enquête [Zhao and Tang, 2020]

L'objectif de la publicité en ligne est d'attribuer les bonnes publicités aux bons utilisateurs afin de maximiser les revenus, le taux de clics (click-through rate CTR en anglais) ou le

retour sur investissement (ROI :return on investment an anglais) de la campagne publicitaire. Les deux principales stratégies marketing de la publicité en ligne sont la livraison garantie (GD :guaranteed delivery) et les enchères en temps réel (RTB :real-time bidding).

Dans le cadre de la GD : les annonces partagent la même idée et le même sujet. Les solutions sont basées sur des algorithmes d'optimisation hors ligne, puis réglées pour une configuration en ligne, ce qui se fait en proposant une approche d'apprentissage multi-facteurs augmenté (MARL: multi-agent reinforcement learning).

RTB permet à un annonceur de soumettre une enchère pour chaque impression individuelle dans un laps de temps très court. Et l'algorithme vise à maximiser la récompense à long terme.

L'apprentissage par renforcement basé sur un modèle est proposé pour apprendre les stratégies d'enchères dans la publicité RTB], où le réseau neuronal est utilisé pour estimer la valeur de l'état, le problème est modélisé comme un problème de contrôle, et RewardNet est conçu SIGWEB pour générer des récompenses afin de résoudre le piège de conception de récompense, au lieu d'utiliser la récompense immédiate. qui prend en compte les enchères des autres annonceurs dans le système, et une approche de clustering est introduite pour résoudre le problème du grand nombre d'annonceurs. [28]

1.10.4. Diriger la recherche exploratoire: apprentissage par renforcement des interactions des utilisateurs avec des mots clés [Głowacka et al., 2013]

Afin d'aider les utilisateurs à diriger l'exploration de l'espace d'informations pendant la recherche, un système de recherche d'informations interactif (RLR :Reinforcement Learning Retrieval) qui applique le RL aux commentaires des utilisateurs sur mots clés a été proposé par [Głowacka et al., 2013].

Le système permet aux utilisateurs d'explorer l'espace d'informations en manipulant les caractéristiques du document (mots-clés) pour indiquer leurs intérêts, et l'apprentissage par renforcement est utilisé pour modéliser l'utilisateur en permettant au système de faire le compromis entre l'exploration et l'exploitation. Cela donne aux utilisateurs la possibilité de diriger plus efficacement leur recherche plus près, plus loin et en suivant une direction. Ces interactions entraînent des prédictions de nouveaux mots-clés et documents correspondant à l'intérêt de l'utilisateur à l'itération actuelle de la récupération.

Le système RLR expose les utilisateurs à un plus grand nombre d'articles pertinents et nouveaux qui ne peuvent pas être facilement trouvés avec un système de recherche simple basé sur des requêtes, mais donne toujours accès aux plus courants.

Lors de l'analyse des performances dans le temps, celle du système RLR n'est augmentée que lorsque RL est utilisé pour créer un modèle utilisateur et que l'utilisateur est en mesure de diriger la recherche plus efficacement, en notant positivement les mots clés recommandés par RLR. De plus, les résultats ont montré que le système RLR offre un meilleur rappel temporel tout en ne sacrifiant pas la précision temporelle, la gardant approximativement la même ou un peu mieux. [29]

1.10.5. Utiliser l'apprentissage par renforcement pour connaître le classement de pertinence des requêtes de recherche [SANDUPATLA 2015]

Une approche triviale de l'utilisation des données de clics pour améliorer le classement des résultats de recherche consiste à organiser les résultats dans l'ordre décroissant du nombre de clics reçus. Ici, l'hypothèse est que, étant donné une paire arbitraire de résultats de recherche, celui qui a la pertinence la plus élevée reçoit plus de clics. Cependant, cette hypothèse n'est pas nécessairement vraie pour la plupart des moteurs de recherche du monde réel, car les clics des utilisateurs sur un résultat dépendent également de la position de ce résultat dans le classement des résultats de recherche. Ainsi, toute méthode qui exploite les données de clics comme commentaires implicites pour améliorer le classement des résultats de recherche doit prendre en compte la comporte.

La plupart des utilisateurs ne cliquent pas au hasard sur les résultats de la recherche. Ils choisissent plutôt délibérément les résultats sur lesquels cliquer après un examen minutieux.

Pour une requête donnée, pour un résultat spécifique, la probabilité de cliquer sur ce résultat diminue avec sa position dans l'ordre de classement. En d'autres termes, la probabilité de clic est non seulement influencée par la pertinence d'un résultat mais également par son classement dans la page de résultats de recherche.

Un modèle pour connaître le classement de pertinence des requêtes de recherche a été proposé par [Sandupatla 2015]. Ce modèle effectue à la fois le processus d'exploration et d'exploitation ensemble selon un facteur Alpha. La valeur de ce facteur fournit la partie du temps que nous avons consacré à l'exploration et le reste du temps est utilisé pour l'exploitation. Cela signifie que le modèle consacre un certain temps à l'exploration même

après avoir acquis les connaissances suffisantes, ce qui diminue l'efficacité du système. La présentation des documents dans chaque processus est exactement similaire au modèle divisé. C'est-à-dire pendant la phase d'exploration. Le succès de ce modèle dépend de l'identification de la meilleure valeur alpha, où Alpha est un nombre binaire dont la valeur est toujours égale à 0 (exploration) ou 1 (exploitation). [30]

1.11. Conclusion

Dans ce chapitre, nous avons présenté le domaine des systèmes de recherche d'information, qui sont devenus omniprésents ces dernières années dans de nombreux domaines.

Ces systèmes sont conçus pour aider les utilisateurs à trouver des documents qu'il de rechercher et répondent le mieux à une question d'un utilisateur grâce à la relation d'appariement. Cette relation s'appuie sur des approches mathématiques.

En se basant sur cette requête, le système de RI exécute une série d'algorithmes qui permettent d'obtenir une liste de documents ordonnés par leur pertinence.

Les approches les plus utilisées dans les SRIs ont été discutées, en mettant l'accent sur les principales méthodes et algorithmes utilisés dans chacune et qui sont liés à notre travail, plus précisément l'apprentissage par renforcement en profondeur en donnant quelques exemples ou travaux qui ont appliqué ces dernières années l'apprentissage en profondeur dans les systèmes de recherche d'information.

Notre proposition sera présentée dans le chapitre suivant.

Chapitre 2 : Conception d'un système de recherche d'information intelligent

2. Introduction

Ce chapitre est consacré à la conception de notre approche de Recherche d'information (RI). On propose dans ce mémoire d'utiliser la notion d'apprentissage profond afin de rendre le SRI adaptatif aux besoins des utilisateurs et qui s'améliore de façon automatique afin d'augmenter sa performance et de satisfaire le mieux les besoins des utilisateurs.

Nous définissons, tout d'abord, la problématique qui donne une petite vision des défis existants que nous sommes entrain de la trouver une solution, ainsi qu'une définition abstraite du système et son architecture avant de passer aux détails de déroulement du processus et l'algorithme proposé.

2.1. Problématique

L'objectif principal des systèmes de Recherche d'Informations (SRI) est permet de retourner à partir d'un ensemble de documents, ceux dont le contenu correspond le mieux à un besoin en informations d'un utilisateur, exprimé à l'aide d'une requête .dont 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 En raison de son efficacité dans le traitement d'énormes ensembles de données (tell que TREC).

Il est rare que la réponse à une requête retournée par le SRI satisfasse l'utilisateur dès le premier essai généralement, en se basant sur cette réponse, l'utilisateur reformule (réécrit) sa requête et la soumet une nouvelle fois au SRI, en espérant avoir un meilleur résultat.

Le retour de pertinence que les utilisateurs fournissent aux systèmes prend généralement place dans un cadre contraignant : les utilisateurs perçoivent le système comme une boîte noire, et ils ne peuvent exprimer l'évolution de leur besoin d'information que sous la forme d'une succession d'évaluations des items reçus. Ainsi, un changement dans les besoins d'information sera potentiellement mal traduit par un utilisateur qui ne reçoit pas du système les documents lui permettant d'exprimer ce changement.

L'utilisateur se trouvant dans l'impossibilité d'adapter son profil à son besoin, finira par abandonner le système, et cela quelle que soit la qualité du système et la qualité d'information qui existe.

De ce point là, la problématique suivante se pose :

Comment peut-on concevoir un système de recherche d'information basé sur l'apprentissage par renforcement profond qui s'adapte aux requêtes de l'utilisateur de manière à mieux répondre à son besoin?

2.2. Objectif et motivation

L'idée derrière la conception de ce système est la mise au point des dernières techniques de l'intelligence artificielle est de développer un système de recherche d'informations intelligent basé sur un apprentissage approfondi et amélioré qui s'adapte aux demandes de l'utilisateur et lui donne les résultats souhaités dès la première tentative.

Comme nous le savons, un SRI est une interface entre les utilisateurs qui expriment leurs besoins par le biais de demandes et d'un large éventail de documents. Un modèle traite ces demandes et renvoie les résultats en fonction des besoins des utilisateurs. Pour ce faire, il intègre un ensemble de fonctions dont les plus importantes sont: le stockage, l'organisation et la sélection des informations qui répondent aux besoins des utilisateurs et enfin la récupération des informations pertinentes c'est-à-dire la recherche de documents dont le contenu correspond au besoin de base des informations de l'utilisateur dans un contexte très large à l'aide de techniques et d'outils permettant d'identifier les informations pertinentes.

Notre système focalise sur la technique l'apprentissage profond, qui est devenu l'un des modèles les plus populaires et plus utilisés qui fonctionnent pour améliorer les capacités de la machine avec ou sans l'aide des humains. On a utilisé le Perceptron multicouche (MLP) pour leur efficacité et performance. Avec une couche d'entrée (requête), une couche de sortie (résultat) et de nombreuses couches cachées qui comprennent plusieurs neurones. Chaque neurone effectue des calculs simples de données d'entrée qui traversent plusieurs couches du compte avant de produire une sortie. Les résultats de la première couche de neurones sont utilisés comme entrées pour calculer la couche suivante et ainsi de suite.

Et à partir de là, plus il y a de couches, plus l'apprentissage de réseaux neuronaux de choses complexes et abstraites est de plus en plus important, et il peut correspondre à la manière de penser humaine. Cependant, il est difficile de développer des mécanismes d'apprentissage efficaces pour les deux classes moyennes (appelées couches profondes ou cachées).

Nous allons essayer:

- D'étudier et comprendre la variabilité des performances des SRI dans le traitement des différentes requêtes.
- D'agir sur la définition des traitements en réalisant une phase d'apprentissage pour trouver le meilleur comportement du SRI.
- D'agir sur la restitution des résultats à l'utilisateur en proposant une visualisation appropriée du résultat fourni par le SRI.
- D'appliquer un apprentissage profond à notre système et essayez de répondre aux demandes des utilisateurs dès la première tentative de recherche.
- Les documents retrouvés par le SRI seront affichés à l'utilisateur afin qu'il sélectionne à son tour l'information pertinente. Le nombre de documents restitués et la longueur de chacun ont une incidence sur la présentation des résultats.

Plus de détails sur les formules utilisées et les calculs des listes d'items de la recherche d'information et l'apprentissage profond seront discutés dans la partie décrivant notre technique.

Dans cette section, nous présenterons notre système de recherche d'informations, en donnant une petite architecture du système ainsi qu'une description formelle de l'environnement. Ensuite, nous expliquons comment l'apprentissage profond affecte l'environnement de travail. Enfin, nous détaillons le processus de recherche d'informations proposé.

2.3. UN SYSTEME DE RECHERCHE D'INFORMATION :

Dans cette section, nous présentons notre système de recherche d'informations adaptatif basé sur l'apprentissage profond. Tout d'abord, nous présentons une architecture qui schématise le processus du fonctionnement de notre système. Ensuite, nous expliquons comment l'apprentissage par renforcement profond affecte l'environnement de travail. Enfin, nous détaillons le processus de recherche d'informations proposé.

2.3.1. Architecture du système

La Figure 2.1 présente l'architecture générale de notre système de recherche d'information et les éléments qui le composent, avec les étapes du processus de la recherche.

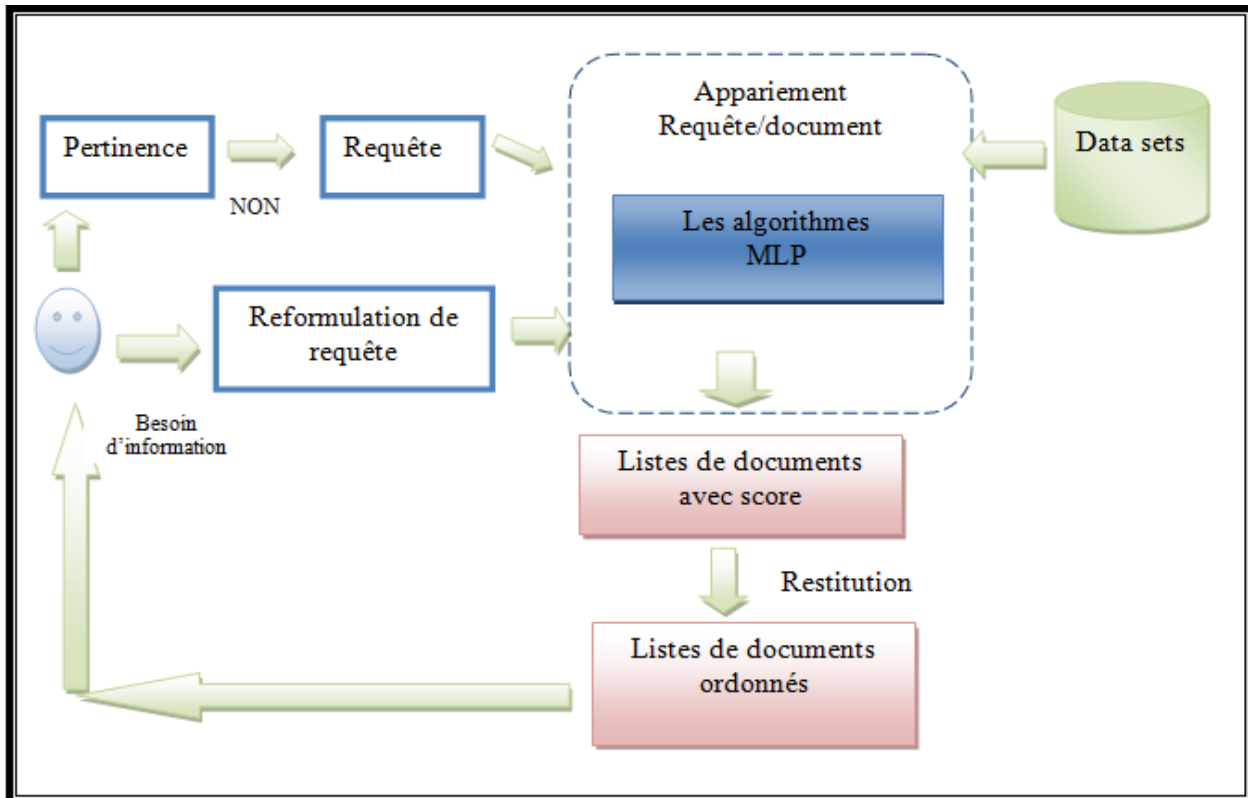


Figure 2.1: Architecture globale du système de recherche d'information

La Figure 2.1 présente l'architecture de notre système de recherche d'information. Le processus de fonctionnement commence par l'interaction de l'utilisateur avec le système en exprimant son besoin en informations sous forme d'une question qui représente la requête. Ensuite, le système calcule la similarité (appariement) entre la requête et les documents de notre base de documents. Le système extrait les données de la collection à travers le modèle de recherche, qui peut interroger la base de données et classe les documents par rapport à un score attribué à chaque document.

Ensuite, le résultat de la recherche de document est ordonné dans un ordre décroissant du rank attribué à chaque document (du document le plus pertinente au moins pertinent par rapport à la requête).

Enfin, si résultat retournés par le système ne répondent pas vraiment au besoin de l'utilisateur, il peut intervenir et reformule la requête pour avoir de meilleur résultat que le premier.

2.3.2. La description du système

En vue de répondre à la problématique posée ci-dessus, on propose dans ce travail, un système de recherche d'information à base de l'apprentissage profond. L'idée de base derrière la conception de notre système est de construire un système capable de répondre

aux besoins des utilisateurs avec des documents pertinents, pour leur faciliter la phase de la recherche et leur aider à choisir et utiliser les bonnes recherches.

Afin d'assurer la bonne recherche de documents pertinents par apport à la requête, on a utilisé les algorithmes de l'apprentissage en profondeur pour améliorer la performance du système et la prédiction des résultats.

Finalement, nous montrons comment le système de recherche d'information sélectionne les documents à l'utilisateur à chaque fois quand on ira en profondeur.

2.3.2.1. La représentation du monde :

Dans cette section, et pour détailler notre technique de façon claire et compréhensible, nous présentons la description formelle de notre approche du DL.

Nous avons utilisé un ensemble de symboles universels afin de les conserver dans toutes les étapes d'explication. Ainsi, nous montrons comment on peut représenter les documents et les utilisateurs et les reformuler à un vecteur qui sert comme entrée pour notre réseau de neurones multicouche.

Nous envisageons un monde avec :

- Un ensemble d'utilisateurs $U = \{u_1, u_2, \dots, u_n\}$;
- Un ensemble de documents $D = \{d_1, d_2, \dots, d_k\}$;
- l'ensemble $T = \{t_1, t_2, \dots, t_n\}$ de tous les termes utilisables pour la rédaction d'un document.
- Un ensemble d'évaluations $R = (p, np)$; où R est l'évaluation de la pertinence de documents par l'utilisateur et qui peut prendre deux formes document pertinente ou non pertinente.

Un utilisateur peut, évalué les résultats du système Les utilisateurs effectuent leurs préférences explicitement en valeurs numériques dans un intervalle de $R = [0,1]$, tel que 0 indique une satisfaction minimale (document non pertinente) et 1 indique une satisfaction maximale (document pertinente).

2.3.2.2. Pré-traitement :

Comme nous travaillons sur des données documentaires, certains processus de pré-traitement sont nécessaires, afin de faciliter les tâches de recherche à savoir la tokenisation, la suppression des ponctuations et des mots vides ..etc

a) Préparation de données :

Dans cette étape, nous allons associer à chaque document une séquence (un vecteur) de 1 ou de 0 selon que le terme t_i est présent ou non dans le document d .

b) La tokenization : On veut dans un premier temps étudier le vocabulaire utilisé dans chaque description de document. On va utiliser la fonction « preprocess »

Le fait d'essayer d'harmoniser les tokens est un processus nommé « normalisation ».

c) Elimination des mots vides

La première manipulation souvent effectuée dans le traitement du texte est la suppression de ce qu'on appelle en anglais les stopwords.

Ces derniers sont des mots très courants dans la langue étudiée ("il", "à", "la"... en français) qui n'apportent pas de valeur informative pour la compréhension du "sens" d'un texte, document et corpus. Il faut donc les supprimer parce qu'ils sont très fréquents et ralentissent le travail. Il existe dans la librairie NLTK une liste par défaut des stopwords dans plusieurs langues. Dans le cadre de ce travail, on supprime d'abord les mots les plus fréquents du corpus puis les stopwords fournis par NLTK.

d) Lemmatisation ou racinisation

Le processus de « lemmatisation » consiste à représenter les mots sous leur forme canonique. Par exemple pour un verbe, ce sera son infinitif. Pour un nom, son masculin singulier. L'idée étant encore une fois de ne conserver que le sens des mots utilisés dans le corpus.

Il existe un autre processus qui exerce une fonction similaire qui s'appelle la racinisation (ou stemming en anglais). Cela consiste à ne conserver que la racine des mots étudiés. L'idée étant de supprimer les suffixes, préfixes des mots afin de ne conserver que leur origine. C'est un procédé plus simple que la lemmatisation et plus rapide à effectuer puisqu'on tronque les mots essentiellement contrairement à la lemmatisation qui nécessite d'utiliser un dictionnaire.

Dans notre cas, on applique une racinisation parce qu'il n'existe pas de fonction de lemmatisation de corpus anglais dans NLTK.

2.3.2.3. Algorithmes utilisés :

Dans cette section nous présentons l'algorithme que nous avons utilisé dans le cadre de ce travail.

a) Perceptron multicouche (*multilayer perceptron* MLP) :

Le réseau neuronal artificiel organisé en plusieurs couches au sein desquelles une information circule de la couche d'entrée vers la couche de sortie uniquement, Chaque couche est constituée d'un nombre variable de neurones, les neurones de la dernière couche (dite « de sortie ») étant les sorties du système global.

➤ Déterminer le nombre de neurones d'entrée et de sortie

Le réseau utilisé dans le cadre de ce travail, possède une couche d'entrée (la requête), une couche de sortie (résultat restitué), et des couches cachées (intermédiaires).

➤ Définir le nombre de couches cachées

Un réseau utilise une couche cachée de neurones pour créer sa propre représentation interne en fonction du problème à résoudre. Cette couche est alors considérée comme un niveau de prétraitement de l'information avant la décision finale. Il est conseillé de respecter la complexité du problème et la nature des données. La détermination de l'architecture du réseau dépend aussi de la capacité de l'ordinateur. Plus un réseau a de couches cachées (donc plus de connexions), plus le traitement nécessite de mémoire et de temps de calcul. Le temps d'apprentissage est proportionnel à la taille de l'échantillon d'apprentissage, la taille du réseau (nombre total de neurones) et le nombre de présentations de l'échantillon d'apprentissage.

➤ Fixer Le nombre de neurones dans la couche cachée

Déterminer le nombre de neurones dans la couche cachée est difficile. Cependant, plus le nombre de couches cachées ou le nombre de neurones dans la couche cachée sont élevés, plus le système apprend pendant l'entraînement et dispose d'un grand nombre de liberté et d'amélioration à partir de différents échantillons d'apprentissage, ce qui conduit à des solutions qui peuvent être différentes.

➤ **Choisir la fonction d'activation :**

Dans notre travail, on considère la fonction \tanh (Tangente hyperbolique)

La fonction \tanh est similaire à la fonction sigmoïde, mais la fonction est comprise entre -1 et 1 :

Son utilisation dans les unités d'un réseau de neurones fonctionne presque toujours beaucoup mieux que l'utilisation de la fonction sigmoïde.

✓ L'équation de la tangente hyperbolique est donnée par :

$$Y = \tanh(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}} \quad (2.1)$$

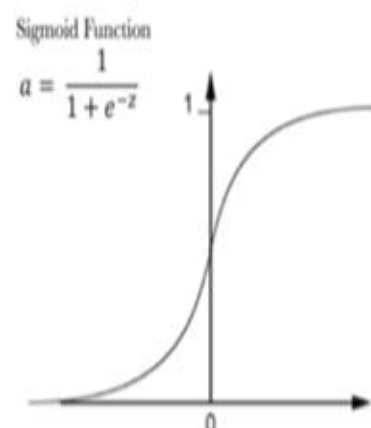
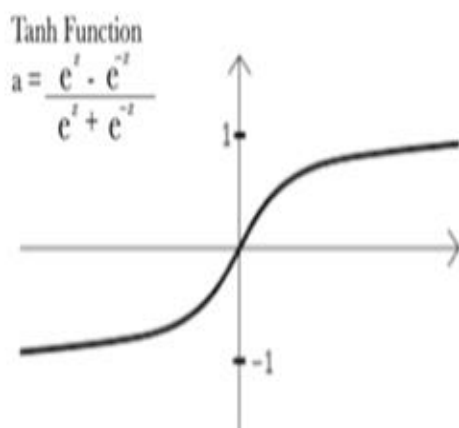
Cette fonction est différentiable et monotone.

✓ La dérivée est donnée par :

$$Y' = 1 - \frac{(e^x - e^{-x})^2}{(e^x + e^{-x})^2} \quad (2.2)$$

✓ La dérivée de la fonction \tanh peut aussi être exprimée en fonction de la fonction \tanh :

$$Y' = 1 - \tanh^2(x) \quad (2.3)$$



- **La fonction Coût :** La fonction de coût la plus utilisée est celle qui minimise l'erreur quadratique, elle peut être écrite :

$$C = \frac{1}{|P|} \sum_{x \in P} \sum_{i \in O} (d_i^x - s_i^x)^2 \quad (2.4)$$

Où P est l'ensemble des exemples d'apprentissage, O l'ensemble des neurones de sortie, s_i^x la valeur du neurone de sortie i après la présentation de l'exemple x et d_i^x la valeur désirée pour le neurone correspondant

- **Architecture du réseau :**

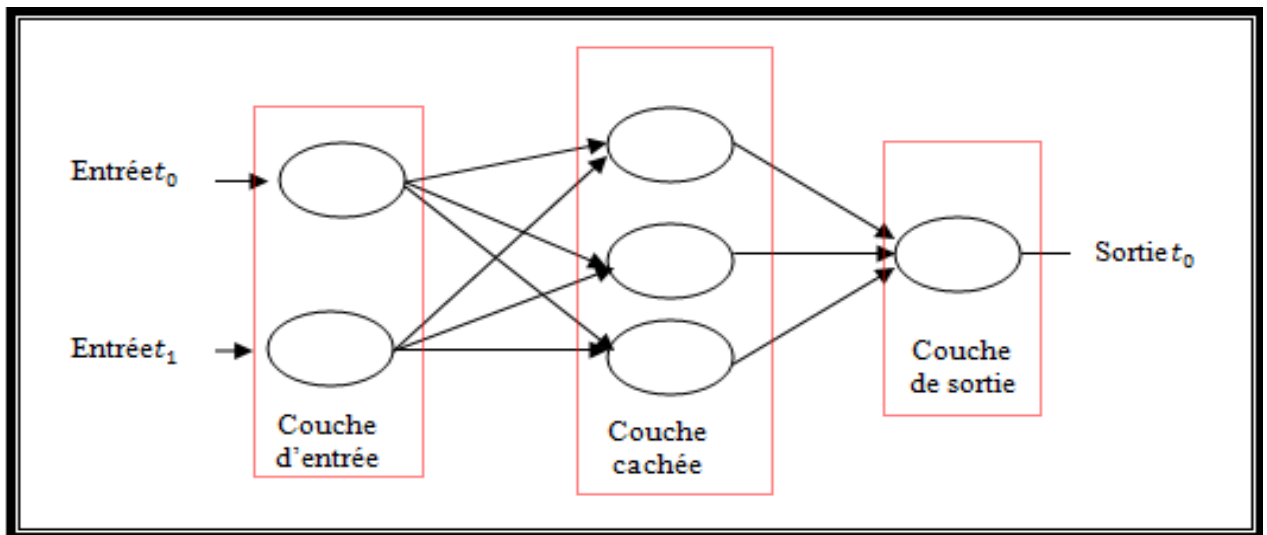


Figure 2.2 : Les couches de MLP

- **Etapes de l'algorithme :**

Les étapes de l'algorithme sont présentées comme suit :

Initialiser les poids du réseau

Initialiser $\Delta w_{kj} \leftarrow 0, \Delta w_{ji} \leftarrow 0$

Répéter jusqu'à terminaison

Pour chaque exemple d'apprentissage faire

① Appliquer le réseau et calculer les sorties

② Calculer et cumuler les deltas

- Pour chaque unité de sortie k
Calculer $\delta_k; \Delta w_{kj} \leftarrow \Delta w_{kj} - \eta \delta_k z_j$
- Pour chaque unité cachée j
Calculer $\delta_j; \Delta w_{ji} \leftarrow \Delta w_{ji} - \eta \delta_j x_i$

Ajuster les poids

$$w_{ji} \leftarrow w_{ji} + \Delta w_{ji}$$

$$w_{kj} \leftarrow w_{kj} + \Delta w_{kj}$$

2.4. Conclusion

Dans ce chapitre, nous avons fixé comme objectif l'utilisation de l'apprentissage en profondeur dans les systèmes de recherche d'information pour obtenir les documents pertinents par rapport à la requête de l'utilisateur afin d'évaluer la performance du système.

Pour cela nous avons proposé un système basé sur l'algorithme MLP.

Les expérimentations de notre système seront présentées dans le chapitre suivant.

Chapitre 3 : Implémentation et Évaluation

1. Introduction

Après avoir présenté l'architecture générale de notre système de recherche d'information, dans ce chapitre, nous allons mener un ensemble d'expériences pour examiner et étudier l'efficacité et la faisabilité des différentes propositions que nous avons présentées dans le chapitre précédent, et les résultats de notre application ainsi qu'une calculer la précision à chaque fois sur TREC dataset pour de mieux évaluer notre application et calculer la performance de système.

1.1. Dataset

Nous avons utilisé dans nos expérimentations la base de données TREC afin d'évaluer notre algorithme.

L'ensemble de données TREC comprend Plus de 16000 documents médicale divisé sur 5 ensembles (breast cancer, healthy, hiv, melanoma, prostate cancer).

thèmes	Nombre de document dans la base TREC
Breast cancer	6716
healthy	4917
hiv	3944
melanoma	1121
Prostate cancer	2265

Tableau 3.1 : nombre de documents de chaque classe dans la base de données

- Nous nous sommes appuyés sur l'étiquette indiquée dans le tableau pendant les expériences

Disease-id	Disease
0	BREAST CANCER
1	HEALTHY
2	HIV
3	MELANOMA
4	PROSTATE CANCER

- Les données sont organisées en documents de discussion différents, Chaque section de la base de données contient de quatre coordonnées différent (disease, title, abstract, nct_id).

1.2.2. Logiciels et Plateformes utilisés

➤ Environnement de travail :



- **Anaconda:** ANACONDA

Anaconda est une distribution libre et open source des langages de programmation python et R appliqué au développement d'applications dédiées à la science des données et à l'apprentissage automatique (traitement de données à grande échelle, analyse prédictive, calcul scientifique), qui vise à simplifier la gestion des paquets et de déploiement. Grâce à cet outil on peut éviter les erreurs d'installation des packages de programmation pour l'apprentissage, que ce soit pour Windows ou linux et il est si simple dans l'utilisation.[W11]

Afin d'implémenter nos modèles, nous avons utilisé plusieurs bibliothèques de python comme pandas ,numpy ,matplotlib ,et les bibliothèques spécialement de l'apprentissage automatique comme :



- **Jupyter :**

Jupyter est une application web utilisée pour programmer dans plus de 40 langages de programmation, dont Python, Julia, Ruby, R, ou encore Scala. Jupyter permet de réaliser des calepins ou notebooks, c'est-à-dire des programmes contenant à la fois du texte en markdown et du code en Julia, Python, R... Ces calepins sont utilisés en science des données pour explorer et analyser des données[W12]

- **Matplotlib** est une bibliothèque complète pour créer des visualisations statiques, animées et interactives en Python, peut être utilisé dans les scripts Python, les serveurs d'applications Web et diverses boîtes à outils d'interface utilisateur graphique. [W13]
- **Sklearn** est un module Python intégrant des algorithmes d'apprentissage automatique classiques dans le monde très uni des packages scientifiques Python (numpy, scipy, matplotlib), Il vise à apporter des solutions simples et efficaces aux problèmes d'apprentissage.[W14]

- **NLTK** est une plate-forme de premier plan pour la création de programmes Python pour travailler avec des données en langage humain. Il fournit des interfaces faciles à utiliser pour plus de 50 corpus et ressources lexicales telles que WordNet, ainsi qu'une suite de bibliothèques de traitement de texte pour la classification, la tokenisation, la recherche de racines, le marquage, l'analyse et le raisonnement sémantique.[W15]
- **Joblib** est un ensemble d'outils pour fournir un pipeline léger en Python, qui permet de sauvegarder des structures de données dans un fichier pour pouvoir les recharger après (par exemple un modèle de machine learning). [W16]

1.3. Objectifs des expérimentations :

1.3.1. Métriques d'évaluation :

❖ Evaluation de la performance du système :

Pour évaluer la performance de notre système de recherche d'information, en termes de réponses (documents restitués) retournées à l'utilisateur, nous avons utilisé les métriques du Rappel, Précision et la métrique F1.

Où, F1 est calculée comme suit :

$$F_1 = \frac{2PR}{P+R} \quad (3.1)$$

Où P et R sont la *Précision* et le *Rappel* respectivement, et ils sont calculés comme suit :

$$P = \frac{N_t}{N} \quad (3.2)$$

$$R = \frac{N_t}{N_p} \quad (3.3)$$

- N_p : Nombre total des items pertinents.
- N : Nombre des items pertinents trouvés.
- N : Nombre total des items

❖ Evaluation de la performance modèle de recherche :

La valeur d'erreur (loss) est égale à la valeur absolue de la valeur réelle moins la valeur prédictive du système

$$\text{Loss} = | \text{val réel} - \text{val prèd} | \quad (3.4)$$

1.3.2. Méthodologie d'implémentation et de test

Afin d'implémenter notre système, nous avons suivies certaines étapes que nous présentons dans ce qui suit :

Etape 1 : télécharger les bibliothèques python nécessaires

```
import pandas as pd
import ast
import joblib
from sklearn.preprocessing import MaxAbsScaler
from sklearn.model_selection import train_test_split
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.neural_network import MLPClassifier
```

Etape 2 : Tout d'abord, nous allons télécharger la base de données (dataset.csv) et commencer à travailler dessus.

```
df = pd.read_csv('dataset.csv')
df = df[['Title', 'Abstract', 'Disease']]
display(df.head())
index = 5500
display(df['Abstract'][index])
df['Disease'][index]
```

	Title	Abstract	Disease
0	Positron Emission Tomography in Monitoring Tre...	This study will investigate the sensitivity a...	Breast Cancer
1	Study of AZD2014 and Palbociclib in Patients W...	This dose finding/extension study consists of...	Breast Cancer
2	Safety and Efficacy Study of Enzalutamide in P...	The purpose of this study is to determine if ...	Breast Cancer
3	A Notch Signalling Pathway Inhibitor for Patie...	An investigational study to determine the saf...	Breast Cancer
4	A Clinical Trial to Compare Efficacy and Toler...	This study will assess the relationship betwe...	Breast Cancer

- Nous nettoyons la base de données et supprimons les mots vides (testing_dataset.csv)

```
def datasetVectorizer(data,target):
    meaningful_words = []
    stop= set(stopwords.words('english'))
    #da[1]

    data.apply(word_tokenize)
    #target.apply(word_tokenize)
    for i in range(len(data)):
        data[i].lower().split()
    meaningful_words = [w for w in data if not w in stop]
    tf = TfidfVectorizer()
    t = tf.fit_transform(meaningful_words)
    t.shape
    return [tf,t,target]
[tf,t,target] = datasetVectorizer(xData,diseases)
```

Le fichier (*dictionary.sav*) contient les différents mots de documents après la suppression des mots vides (stop words).

Etape 3 : Division de données.

On doit trier les données et on les divise en deux : 60% pour l'apprentissage et 40% pour le test.

```
x_train,x_test,y_train,y_test = train_test_split(t,target,test_size = 0.4)
```

Etape 4 : formation du modèle

Nous avons formé trois modèles différents tout en changeant le nombre de couches cachées à chaque fois (8, 16, 24) et nombre d'itérations fixé= 1000 et les résultats dans les fichiers (trainedModel.sav).

```
: ctivationFunction = 'tanh'
x_train,x_test,y_train,y_test] = joblib.load('trainxy.sav')
lf = MLPClassifier(hidden_layer_sizes = hiddenLayer, activation = activationFunction,learning_rate = 'adaptive', max_iter = 1000)
lf.fit(x_train,y_train)
oblib.dump([clf,tf],modelFileName)

: ['trainedModel_8_2.sav']
```

Etape 5 : évaluation et test

❖ But des tests :

Nous menons une série d'expériences pour étudier l'efficacité et la faisabilité des différentes propositions faites dans le chapitre précédent.

Ces expériences sont menées pour trouver les meilleures valeurs pour chaque variable avec le taux d'erreur le plus faible en fonction des tentatives faites par le système à chaque apprentissage en changeant le nombre de couches cachées et en observant la réponse du système à chaque fois.

Nous voulons aussi la prévisibilité du système et sa capacité à classer une phrase pour l'une des catégories en fonction d'une phrase pour obtenir un bon résultat, nous devons faire plusieurs expériences, nous abordons donc les questions suivantes :

- ✓ L'apprentissage en profondeur est-il utile pour les systèmes de recherche d'informations ?
- ✓ Quels sont les meilleurs critères pour trouver un meilleur résultat ?
- ✓ L'ajout du nombre de couches cachées peut-il affecter les résultats obtenus ?
- ✓ Quelles sont les meilleures attentes du système pour donner un résultat plus proche de ce que veut l'utilisateur ?

1.4. Expérimentations et Tests:

Nous présentons dans ce qui suit la série d'expérimentations que nous avons fait afin d'évaluer notre SRI :

1.4.1. Expérimentation 1 : (Couches Cachées = 8)

Dans cette expérience, nous avons fixé le nombre de couches cachées à 8 couches et le nombre d'itérations maximal $maxiter=1000$.

- **Performance modèle :**

Les différentes valeurs du pourcentage d'erreur (**Loss Function**) par rapport au nombre d'itérations sont représentées dans la Figure 3.2.

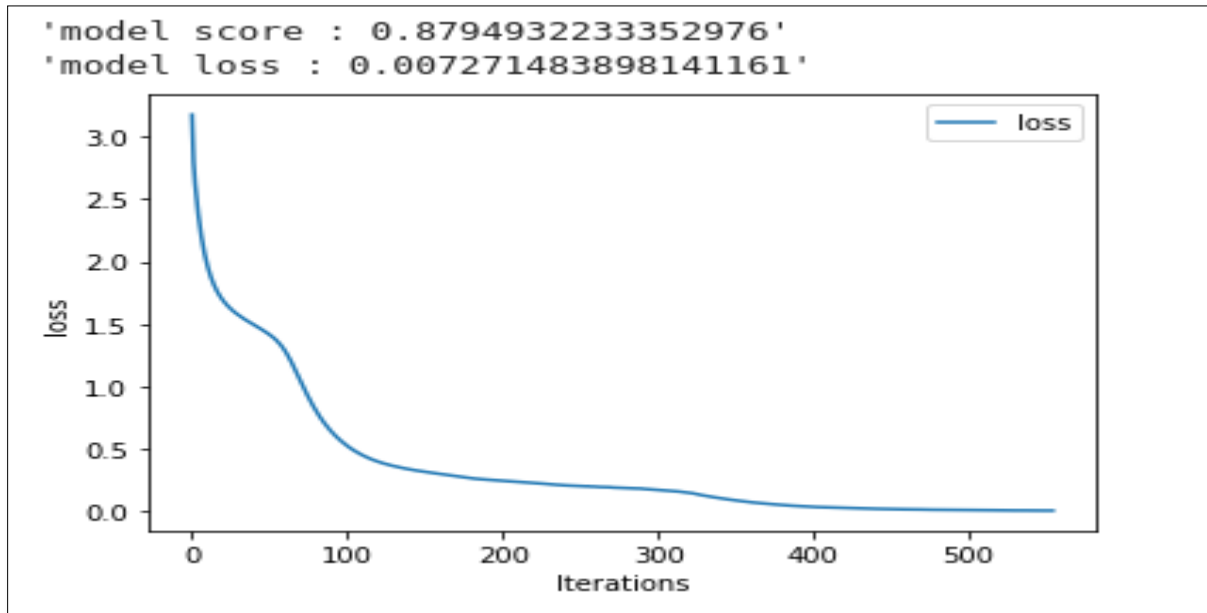


Figure 3.2 : La courbe *loss* avec 8 couches cachées

A partir de la Figure 3.2., nous remarquons que les valeurs de la fonction de perte (Loss Function) diminuent continûment au fur et à mesure que le nombre d'itérations augmente jusqu'à atteindre une valeur minimale de 0,0072714838 et score =0.879493223335.

Et avec ces résultats, nous ne pouvons pas les considérer comme bons tant que nous n'aurons pas mené plus d'expériences.

- **Performance Système :** On calcule la précision, le rappel, métrique F1 ainsi que la macro-moyenne et la moyenne pondérée pour évaluer de chaque classe. Les résultats sont représentés comme suit :

	precision	recall	f1-score	support
0	0.95	0.93	0.94	1853
1	0.96	0.94	0.95	1992
2	0.91	0.96	0.93	1573
3	0.86	0.89	0.88	438
4	0.95	0.93	0.94	932
accuracy			0.94	6788
macro avg	0.93	0.93	0.93	6788
weighted avg	0.94	0.94	0.94	6788

Figure 3.3 : présente les résultats du Rappel et de la précision.

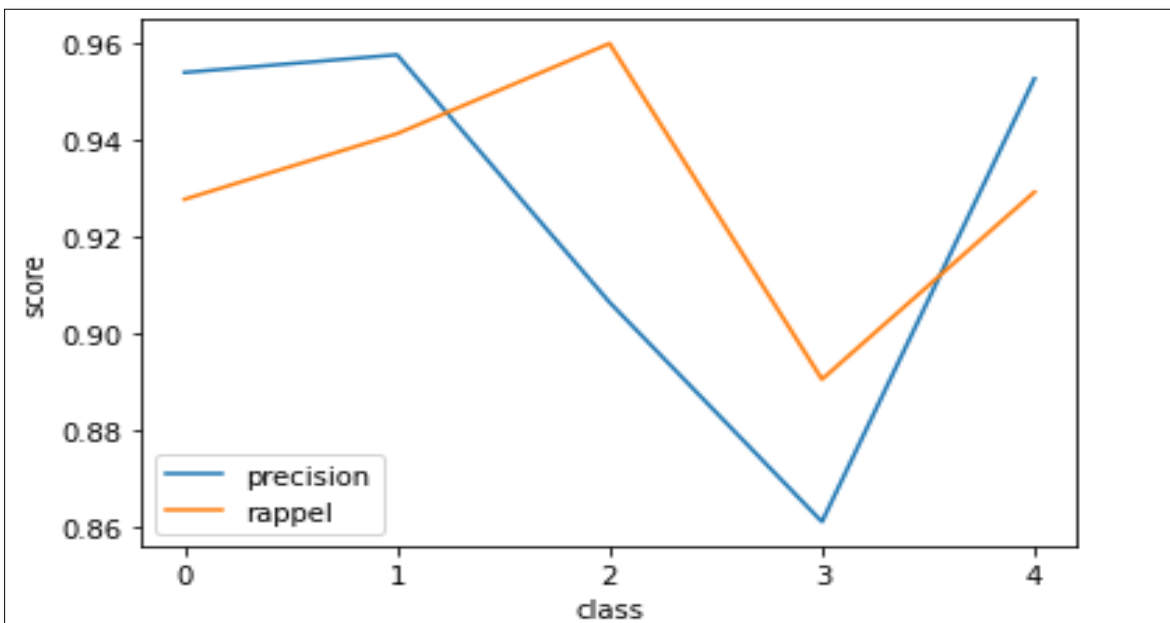


Figure 3.4 : Courbe de Rappel et Précision avec couche cachée =8

Dans la Figure 3.3. On remarque que les valeurs de précision sont similaires dans toutes les catégories, mais elles enregistrent le niveau le plus bas avec *melanoma* avec une valeur = **0.86**. Alors que le rappel a marqué la plus grande valeur avec *hiv* (**0.96**) et la valeur la plus basse avec *melanoma* (**0.89**).

1.4.2. Expérimentation 2 : Couches Cachées = 16

Dans cette expérience, nous avons donné des valeurs constantes pour *maxiter* = 1000 et nombre de couches cachées = 16.

- **Performance modèle :**

L'évolution de la fonction d'erreur (**Loss Function**) par rapport au nombre d'itérations sont présentées dans la Figure 3.4.

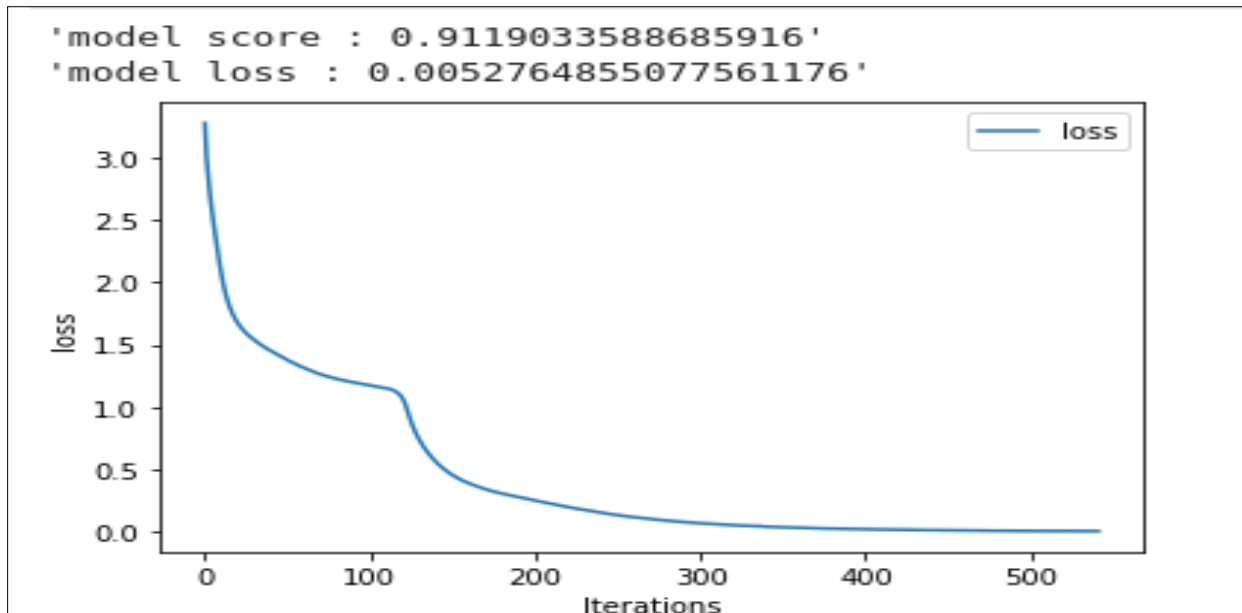


Figure3.4: courbe de la fonction de perte avec 16 couches cachées

La Figure 3.4. montre une légère amélioration des valeurs de la fonction de perte à mesure que le nombre d'itérations augmente, jusqu'à ce qu'elle atteigne une valeur minimale de 0.0052764855 et de score= 0.91190335 (C'est une bonne valeur pour le système) en examinant les chiffres.

Par conséquent, on peut considérer que l'augmentation des couches cachées améliore considérablement les performances du système.

- **Performance Système :**

Les résultats de performance du système Rappel et Précision pour un nombre de couches cachées = 16 sont présentés dans la Figure 3.5.

	precision	recall	f1-score	support
0	0.94	0.92	0.93	1853
1	0.97	0.95	0.96	1992
2	0.91	0.94	0.93	1573
3	0.84	0.89	0.87	438
4	0.93	0.94	0.93	932
accuracy			0.93	6788
macro avg	0.92	0.93	0.92	6788
weighted avg	0.93	0.93	0.93	6788

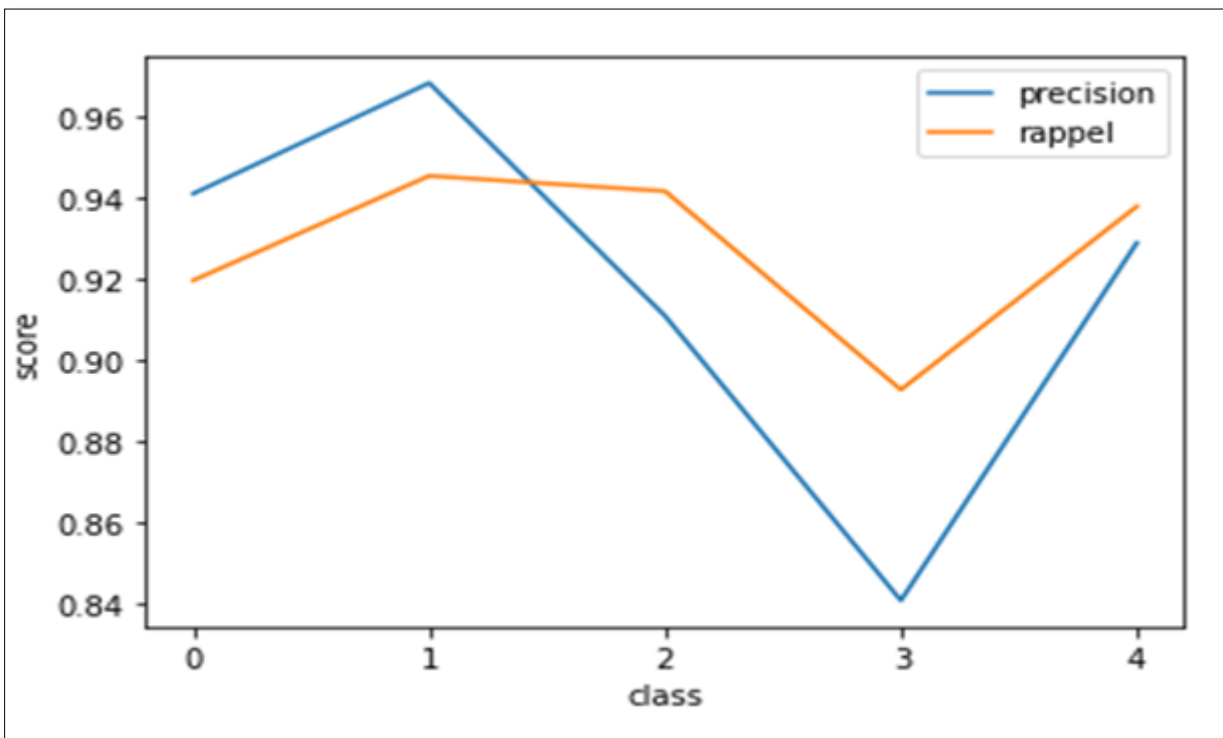


Figure3.5 : Courbe de Rappel et Précision avec 16 couches cachées

A partir de la Figure 3.5 on constate que les valeurs de rappel sont similaires dans toutes les catégories, mais elles enregistrent le niveau le plus bas avec *melanoma* avec une valeur égale à **0.84**. Quant aux valeurs de précision, nous enregistrons la valeur la plus élevée avec *healthy*= **0,97** et le niveau le plus bas avec *melanoma* = **0,84**.

1.4.3. Expérimentation 3 : Couches Cachées = 24

Dans cette expérience, nous avons fixé le nombre d'itérations *maxiter* à 1000 itérations et

le nombre de couches cachées à 24.

- **Performance modèle :**

L'évolution de la fonction de perte (**Loss Function**) par rapport au nombre d'itérations sont présentées dans la Figure 3.6.

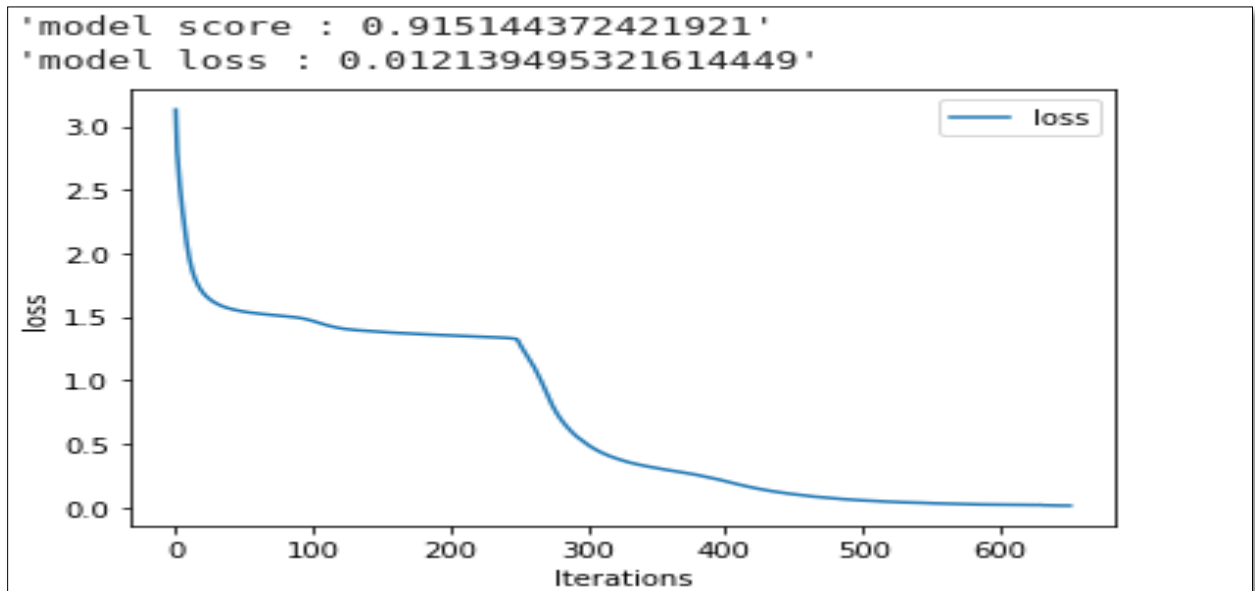


Figure 3.6 : Courbe de la fonction de perte avec 24 couches cachées

La courbe de la Figure 3.6 montre une augmentation des valeurs de perte avec l'augmentation du nombre d'itérations jusqu'à ce qu'il atteigne une valeur égale à 0.012139495 pour 600 itérations, au-delà de ça nous n'obtenons pas d'amélioration des résultats.

- **Performance Système :**

Les résultats de performance du système Rappel et Précision pour un nombre de couches cachées = 24 sont présentés dans la Figure 3.7.

	precision	recall	f1-score	support
0	0.94	0.89	0.92	1853
1	0.92	0.93	0.92	1992
2	0.91	0.95	0.93	1573
3	0.80	0.92	0.86	438
4	0.95	0.90	0.92	932
accuracy			0.92	6788
macro avg	0.90	0.92	0.91	6788
weighted avg	0.92	0.92	0.92	6788

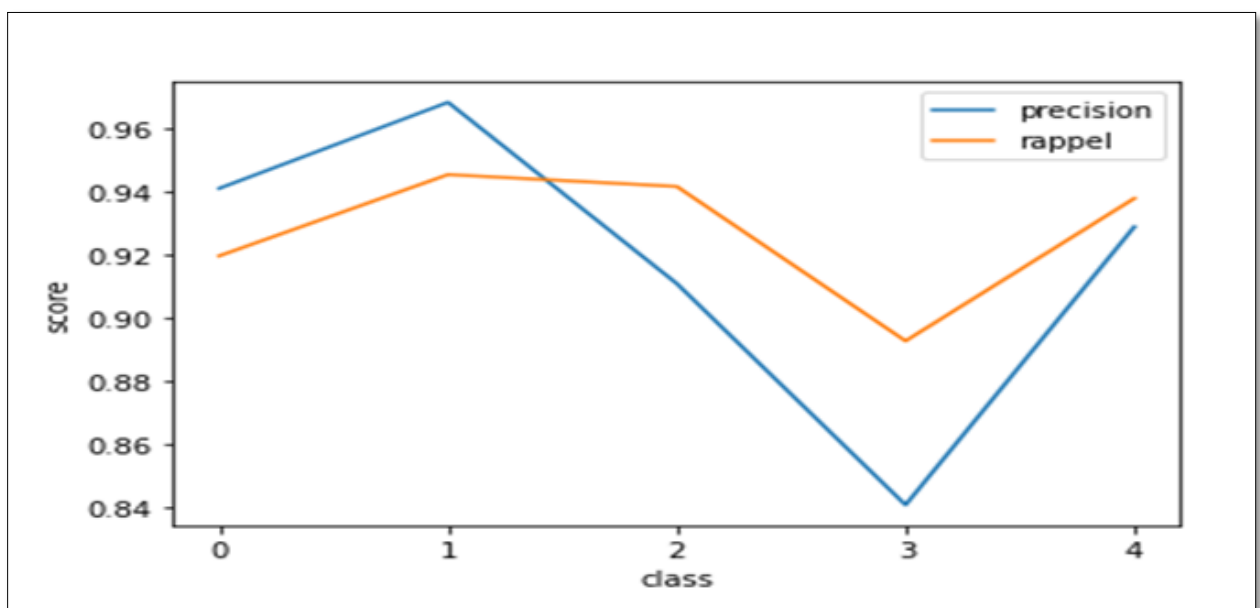


Figure 3.7 : Courbes de Rappel et Précision avec 24 couches cachées

De la Figure 3.7 on constate que les valeurs de Précision sont similaires dans toutes les catégories, mais elles enregistrent le niveau le plus bas avec *melanoma* avec une valeur égale à **0.80** et enregistrons la valeur la plus élevée avec *prostate cancer* = **0,95**. Quant aux valeurs de Rappel, nous enregistrons la valeur la plus élevée avec *hiv* = 0,95 et le niveau le plus bas avec *prostate cancer* = 0,89.

On peut constater que les valeurs de Précision et du Rappel changent dans toutes les classes avec l'augmentation du nombre de couches cachées du réseau.

Partie 2 : Obtenez une prédiction pour la maladie

Dans cette partie, nous faisons des prédictions pour le système en entrant quelques requêtes de test et le système classant dans l'une des cinq catégories de l'ensemble de données (HIV, melanoma, healthy, breast cancer, prostate cancer)

Observation

Les mots utilisés dans la requête ne sont pas explicitement présents dans l'ensemble de données et le système est classé par la fonction d'intelligence artificielle dès qu'il fait allusion à l'une des maladies qu'il existe dans la base. Mais si la phrase ne suggère pas ce qui est requis, le système ne donne aucune réponse.

Experimentation 1:

Nous avons entré la phrase (requête 1) suivante "to evaluate the effect of early viral reactivation by latency reversing agents (LRA) and/or administration of potent broadly neutralizing antibodies" comme entrée

Et les attentes du système étaient de donner le résultat "hiv" comme une sortie

```
[17]: ing = 'To evaluate the effect of early viral reactivation by latency reversing agents (LRA) and/or administration of potent broadly neutralizing antibodies'

Disease(entredString):
,tf] = joblib.load(modelFileName)
= tf.transform([entredString])
= clf.predict_proba(vec)[0]
ax(pre)>0.2:
pre = pre.tolist().index(max(pre))
pre = diseases1[pre]
:
pre = ''
rn pre

ase(inputString)

[17]: 'HIV'
```

Figure 3.8 : résultat de prédiction avec requête 1

Experimentation 2:

Nous avons entré la phrase (requête 2) suivante "the primary objective of the study was to assess the bioequivalence of the oral suspension formulation the oral solution formulation" comme entrée

Et les attentes du système étaient de donner le résultat "healthy" comme une sortie

```
[10]: inputString = 'The primary objective of the study was to assess the bioequivalence of the oral suspension formulation the oral solution formulation and the  
  
def findDisease(entredString):  
    [clf,tf] = joblib.load(modelFileName)  
    vec = tf.transform([entredString])  
    pre = clf.predict_proba(vec)[0]  
    if max(pre)>0.2:  
        pre = pre.tolist().index(max(pre))  
        pre = diseases1[pre]  
    else :  
        pre = ''  
    return pre  
  
findDisease(inputString)  
  
[10]: 'HEALTHY'
```

Figure 3.9 : résultat de prédiction avec requête 2

Expérimentation 3 :

Nous avons utilisé la requête 3 suivante "*the exact cause of all melanomas isn t clear ,but exposure to ultraviolet radiation from sunlight or banning lamps and beds*" comme entrée

Et les attentes du système étaient de donner le résultat "prostate cancer " comme une sortie :

```
: tString = 'The exact cause of all melanomas isn t clear, but exposure to ultraviolet radiation from sunlight or tanning lamps and beds  
  
findDisease(entredString):  
[clf,tf] = joblib.load(modelFileName)  
vec = tf.transform([entredString])  
pre = clf.predict_proba(vec)[0]  
if max(pre)>0.2:  
    pre = pre.tolist().index(max(pre))  
    pre = diseases1[pre]  
else :  
    pre = ''  
return pre  
  
Disease(inputString)  
  
: 'PROSTATE CANCER'
```

Figure 3.10 : résultat de prédiction avec requête 3

Experimentation 4:

Nous avons entré la phrase suivante (requête 4) "look for anything new, changing or unusual on both sun-exposed and sun-protected areas of the body melanomas comonly.." comme entrée

Et les attentes du système étaient de donner le résultat " " comme une sortie (n'est existe pas un sortie)

```
7]: inputString = 'Look for anything new, changing or unusual on both sun-exposed and sun-protected areas of the body. Melanomas commonly ap

def findDisease(entredString):
    [clf,tf] = joblib.load(modelFileName)
    vec = tf.transform([entredString])
    pre = clf.predict_proba(vec)[0]
    if max(pre)>0.2:
        pre = pre.tolist().index(max(pre))
        pre = diseases1[pre]
    else :
        pre = ''
    return pre

findDisease(inputString)

7]: ''
```

Figure 3.11: résultat de prédiction avec requête 4

On peut voir que toutes les prédictions du système étaient correctes car nous connaissons déjà la réponse et c'est pourquoi elle peut être considérée comme bonne pour la classification.

1.5. Conclusion:

D'après les différentes expérimentations réalisées dans le cadre de ce travail, nous pouvons conclure que le deep learning est une méthode efficace.

Cependant, il peut être amélioré en le combinant avec d'autres méthodes. De plus, l'utilisation de la méthode MLP nécessite la modification de nombreux paramètres. Ainsi, le bon choix des paramètres peut améliorer les résultats.

Enfin, l'ajout de nombreuses couches cachées dans l'algorithme MLP et l'augmentation du nombre d'itération améliore considérablement la qualité des prédictions.

Conclusion Générale et Perspectives

Les systèmes de recherche d'informations ont été d'une grande importance dans divers domaines pour aider les utilisateurs à choisir les documents qui correspondent à leurs besoins. Ce document vise à présenter une étude bibliographique sur les systèmes de recherche d'informations basée sur les demandes des utilisateurs avec un accent particulier sur les méthodes qui utilisent spécifiquement l'apprentissage automatique, l'apprentissage profond et le calcul des RP.

Dans un premier temps, nous avons discuté des généralités sur les systèmes de recherche d'informations, un domaine en constante évolution et de plus en plus utilisé dans la vie de tous les jours. Nous avons abordé les concepts de base liés à ces systèmes, nous avons discuté des échelles d'évaluation de ces systèmes, et répertorié certains des cadres utilisés dans ce domaine de recherche, et enfin, nous avons vu quelques méthodes de récupération des informations fournies par les chercheurs, dans le but de comprendre une idée générale de l'apprentissage profond, qui est cette structure et l'utilisation de cette dernière dans Des systèmes de recherche d'informations selon leur utilisation et ces différents modèles qui nous ont permis de voir l'apport de ces algorithmes.

Ces dernières années, les systèmes de recherche d'informations ont connu un grand développement, en particulier avec l'avènement du Big Data et le développement d'algorithmes d'apprentissage, les systèmes sont devenus plus complexes et précis. Mais comme nous l'avons vu, la précision du système n'est pas suffisante pour en faire un système puissant. Vous devez vous soucier d'autres métriques comme le rappel par exemple. De nombreux chercheurs se concentrent désormais sur la satisfaction des utilisateurs, la précision n'est plus la seule mesure à améliorer, et la tendance ces dernières années est d'obtenir des systèmes de récupération d'informations qui utilisent le deep learning, qui répond aux demandes des utilisateurs et augmente les performances du système ainsi que pour gérer les problèmes de big data (large base).

Par conséquent, avec la croissance continue du volume de données, RIS est devenu dépendant du mécanisme de recherche pour effectuer la tâche de recherche d'informations et pour satisfaire le besoin de l'utilisateur pour les informations qui sont exprimées à travers la demande.

Par conséquent, le système doit restituer à l'utilisateur le nombre maximum de citations de documents pertinents pour la demande et le nombre minimum de citations de documents non liés.

Références

A. Références Bibliographiques

- [1] Yaël Champclaux , “Un modèle de recherche d’information basé sur les graphes et les similarités structurelles pour l’amélioration du processus de recherche d’information”, Informatique [cs], Université Paul Sabatier - Toulouse III, 2009. Français. fftel-00446372f
- [2] Indexation et Recherche d’Information vidéo, Introduction à la RI ,Modèles de RI
- [3] ,Hannech.A , “Système de recherche d’information étendue basé sur une projection multi-espaces”, UNIVERSITÉ DU QUEBEC A CHICOUTIMI, thèse de Doctorat , Juillet 2018.
- [4] Karen Sauvagnat, “**Modèle flexible pour la Recherche d’Information dans des corpus de documents semi-structurés**”, 2005
- [5] Salton.Gerard, “ Automatic text processing: the transformation, analysis, and retrieval of information by computer”, Addison-Wesley, 1989, 0-201-12227-8
- [6] Amimer.h, Chakroun.F, “Réalisation d’un système de recherche d’information” ,université Abou Berkr Belkaid ,tlemcen ,2015.
- [7] Reddahi Nabil1 , Labriji Amine1 , Abdelbaki Issam1 , Ben Lahmar El Habib1 , Labriji Elhoussine1, “Études des méthodes d’indexation sémantique”, Université Hassan II- Casablanca,2016.
- [8] Bruno Bachimont,“L’indexation multimédia” , chapitre 5.pfd
- [9] Stéphane Crozat, ”Indexation de documents” ,01 septembre 2016
- [10] AMRANE Abdesala, “Vers un langage de haut niveau pour la recherche d’information multimédia”, Université Abderrahmane Mira ,Bejaia, Mémoire de Magister,2013.
- [11] H. Aliane, Z. Alimazighi , R. O. Boughacha, , T. Djelliout , “Un système de reformulation de requêtes pour la recherche d’information”, Université des Sciences et de la Technologie Houari Boumediene, Alger . RIST Vol, 14 n°01 Année 2004.
- [12] Karen Pinel-Sauvagnat, Josiane Mothe , “Mesures de la qualité des systèmes de recherche d’information”, université Toulouse, 2013

- [13] Sajad Mousavi, Enda Howley, Michael Schukat, “Deep Reinforcement Learning: An Overview”, Conference Paper · June 2018, DOI: 10.1007/978-3-319-56991-8_32
- [14] “L’intelligence artificielle définition - généralités- historique-domaines”, Université Paris 5 - Maîtrise de mathématiques - Maîtrise MASS - MST ISASH, Module INTELLIGENCE ARTIFICIELLE ,2000.
- [15] Ioan Landau, Vlad Landau , “Data Mining et Machine Learning dans les Big Data *Une tentative de démystification* ”, GIPSA –LAB, Grenoble, 3 Novembre 2016.
- [16] Aurélien.Garivier ,“ Big data ,Machine learning: qu’est-ce que la science des données ? ”, Journée de l’IREM, Bordeaux, Institut de Mathématiques de Toulouse,LabeX CIMI Université Paul Sabatier,Janvier 2017.
- [17] Boughaba.M, Boukhris.B, “ L’apprentissage profond (Deep Learning) pour la classification et la recherche d’images par le contenu”, Université Kasdi Merbah Ouargla, Mémoire Master Professionnel,2017.
- [18] Dino S. Ratcliffe , Sam Devlin, Udo Kruschwitz , Luca Citi , “ Clyde :A deep reinforcement learning DOOM plying agent”,2017.
- [19] John Schulman, “Deep Reinforcement Learning”, MLSS, May 2016, Cadiz
- [20] David Silver, Google DeepMind, , “ Deep Reinforcement Learning”.
- [21] Vincent François-Lavet, Peter Henderson, Riashat Islam, Marc G. Bellemare ,Joelle Pineau (2018), “An Introduction to Deep Reinforcement Learning”, Foundations and Trends in Machine Learning: Vol. 11, No. 3-4. DOI: 10.1561/22000000071.
- [22] François de La Bourdonnaye, Céline Teulière,,Jochen Triesch, Thierry Château, “Apprentissage par profond de la fixation binoculaire en utilisant de la détection d’anomalies” ,Institut Pascal, CNRS UMR 6602, Aubière, France, Frankfurt Institute for Advanced Studies, Germany.
- [23] Yunshu Du, “improving deep reinforcement learning via transfer”, university Pullmùan ,Wa , Doctoral consortium,Canada ,may 2019.
- [24] “Le problème d’apprentissage par renforcement”.
- [25] Meng Fang ,Yuan Li ,Trevor Cohn, “Learning how to Active Learn: A Deep Reinforcement Learning Approach”, School of Computing and Information Systems, The University of Melbourne .
- [26] Young-Woo Seo,Byoung- Tak Zhang , “A Reinforcement Learning Agent for Personalized Information Filtering”, 1~1 2000 New Orleans LA USA
copyright ACM 2000 1-581 i3-134-8~00~

- [27] Corby Rosset, Damien Jose, Gargi Ghosh, Bhaskar Mitra, and Saurabh Tiwary
Microsoft AI & Research , “Optimizing Query Evaluations Using Reinforcement Learning for Web Search”.
- [28] Xiangyu Zhao, Jiliang Tang , “Deep Reinforcement Learning for Search, Recommendation, and Online Advertising: A Survey”, Michigan State University, arXiv:1812.07127v5 [cs.IR] 17 Jan 2020.
- [29] Dorota Glowacka, Tuukka Ruotsalo, Ksenia Konyushkova , Kumaripaba .Athukorala, Samuel Kaski, Giulio Jacucci , “Directing exploratory search : Reinforcement learning from user interaction with keywords”
- [30] HAREESH SADUPATLA :”USING REINFORCEMENT LEARNING TO LEARN RELEVANCE RANKING OF SEARCH QUERIES”, Purdue university Indianapolis, Indiana, 2016

B. Références Web (Techniques)

[w1]	https://www.google.com/search?q=problematique+de++la+recherche+d%27information&tbm=isch&ved=2ahUKEwjB6fjnoPHrAhVBwOAKHWIDB9wQ2-cCegQIABAA&sq=problematique+de++la+recherche+d%27information&gs_lcp=CgNpbWcQA1DsJ1j9Q2CyTWgAcAB4AIABnQKIAekDkgEDMi0ymAEAo.AEBqgELZ3dzLXdpei1pbWfAAQE&sclient=img&ei=heVjX4GTNsGAwfp3g.DQ&bih=576&biw=1366&rlz=1C1CHBD_frDZ883DZ883#imgrc=RUAoGw6B51-EhM
[w2]	https://www.google.com/search?q=processus+de+recherche+d%27information&rlz=1C1CHBD_frDZ883DZ883&sxsrf=ALeKk018yRFGjH9o_eIAyZRM9Cbn4OHUPA:1583619457414&tbm=isch&source=iu&ictx=1&fir=pUOCVVfy8KuBZM%253A%252C_JX_bEQsF9BRBM%252C_&vet=1&usg=AI4_-kR-VOC_C-IbnG3tWNpJOMIxNU3Q5Q&sa=X&ved=2ahUKEwiMrN-ZsonoAhWOz4UKHwMCDkIQ9QEwAHoECAoQHQ#imgrc=pUOCVVfy8KuBZM
[w3]	https://link.springer.com/referenceworkentry/10.1007%2F978-0-387-39940-9_921
[w4]	https://www.google.com/search?q=Processus%2Bde%2BRI%2BChapitre%2B_%2B1%2Bdocuments%2BLangage%2Bde%2BRequ%2C3%AAtes%2BRequ%2C3%AAtes%2BSRI&rlz=1C1CHBDfrDZ883DZ883&sxsrf=ALeKk01DeRtxd2tvjF9uZwXr7INlq7ZDOW:1600525528797&source=lnms&tbm=isch&sa=X&ved=2ahUKEwiB2uOctvXrAhVTA2MBHegNAR8Q_AUoAnoECAsQBA#imgrc=HdgmR48qzj57iM
[w5]	https://www.google.com/search?q=Processus%2Bde%2BRI%2BChapitre%2B_%2B1%2Bdocuments%2BLangage%2Bde%2BRequ%2C3%AAtes%2BRequ%2C3%AAtes%2BSRI&rlz=1C1CHBDfrDZ883DZ883&sxsrf=ALeKk01DeRtxd2tvjF9uZwXr7INlq7ZDOW:1600525528797&source=lnms&tbm=isch&sa=X&ved=2ahUKEwiB2uOctvXrAhVTA2MBHegNAR8Q_AUoAnoECAsQBA#imgrc=HdgmR48qzj57iM

[w6]	https://www.google.com/search?q=1%E2%80%99apprentissage+par+renforcement&rlz=1C1CHBDfrDZ883DZ883&sxsrf=ALeKk00SjSNGCm7ool4zk1thaQ6pm34KGw:1600440770077&source=lnms&tbm=isch&sa=X&ved=2ahUKEwjAgta8-vLrAhU_A2MBHZ5EA6UQ_AUoAXoECBYQAw&biw=1366&bih=576#imgrc=mQZ0fccKYddKpM.
[w7]	https://fr.mathworks.com/solutions/deep-learning/convolutional-neural-network.html
[w8]	https://datasciencetoday.net/index.php/fr/machine-learning/148-reseaux-neuronaux-recurrents-et-lstm
[w9]	https://www.journaldunet.com/solutions/dsi/1209776-deep-reinforcement-learning-l-ia-capable-de-surpasser-l-humain/
[w10]	https://fr.mathworks.com/solutions/deep-learning/convolutional-neural-network/_jcr_content/mainParsys/band_copy_copy_14735_1026954091/mainParsys/columns_16065_42234_c/2/image.adapt.1200.high.jpg/1595600919678.jpg
[w11]	https://intelligence-artificielle.agency/anaconda/
[w12]	https://www.smartnskilled.com/produit/produit-jupyter-notebook
[w13]	https://pypi.org/project/matplotlib/
[w14]	https://www.kite.com/python/docs/sklearn
[w15]	https://www.nltk.org/
[w16]	http://www.python-simple.com/python-autres-modules-non-standards/joblib.php

Résumé

Les systèmes de recherche d'informations visent à fournir un certain nombre d'informations présentant un intérêt potentiel pour l'utilisateur et correspondant à ses préférences. Cependant, ce type de système souffre de certains problèmes tels que l'expression correcte de la demande et l'obtention des informations les plus appropriées et pertinentes. Avec l'émergence de l'apprentissage automatique ces derniers temps, en particulier l'apprentissage profond, qui est devenu largement utilisé dans de nombreux domaines tels que l'intelligence artificielle, et en raison des progrès et de l'énorme popularité qu'il a atteint, les chercheurs l'ont adopté comme solution aux problèmes des systèmes de recherche d'informations.

Dans ce travail, nous avons proposé un système de recherche d'information adaptatif aux besoins des utilisateurs basé sur l'apprentissage profond. Deux modèles d'apprentissage profond ont été élaborés afin d'améliorer la qualité des réponses du système, sachant l'algorithme MLP et ont montré une amélioration des performances du système et de la qualité des résultats retournés.

Mots clés: *Systèmes de Recherche d'Informations, Apprentissage Automatique, Apprentissage Profond, algorithme MLP, Requête, Pertinence.*

Abstract

Information retrieval systems aim to provide a certain amount of information of potential interest to the user and corresponding to his preferences. However, this type of system suffers from certain problems such as the correct expression of the request and obtaining the most appropriate and relevant information. With the emergence of machine learning in recent times, especially deep learning, which has become widely used in many fields such as artificial intelligence, and due to the advancements and huge popularity that it reached, researchers adopted it as a solution to the problems of information retrieval systems.

In this work, we have proposed an information retrieval system adaptive to user needs based on deep learning. Two deep learning models have been developed to improve the quality of system responses, knowing the mlp algorithm. and showed improvement in system performance and the quality of returned results.

Keywords: *Information Retrieval Systems, Machine Learning, Deep Learning, MLP algorithm, Query, Relevance.*

المخلص:

تهدف أنظمة استرجاع المعلومات إلى توفير قدر معين من المعلومات التي قد تهتم المستخدم وتتوافق مع تفضيلاته. ومع ذلك، فإن هذا النوع من النظام يعاني من مشاكل معينة مثل التعبير الصحيح عن الطلب والحصول على المعلومات الأكثر ملاءمة وذات الصلة. مع ظهور التعلم الآلي في الآونة الأخيرة وخاصة التعلم العميق الذي أصبح يستخدم على نطاق واسع في العديد من المجالات مثل الذكاء الاصطناعي، ونتيجة للتقدم والشعبية الهائلة التي يتمتع بها التي تم التوصل إليها، اعتمدها الباحثون كحل لمشاكل نظم استرجاع المعلومات. في هذا العمل، اقترحنا نظامًا لاسترجاع المعلومات يتكيف مع احتياجات المستخدم بناءً على التعلم العميق. تم تطوير نموذجين للتعلم العميق لتحسين جودة استجابات النظام، مع معرفة خوارزمية MLP. وأظهر تحسناً في أداء النظام وجودة النتائج المرتجعة. الكلمات الرئيسية: نظم استرجاع المعلومات، التعلم الآلي، التعلم العميق، خوارزمية mlp، الاستعلام، الصلة