

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mouloud Mammeri de Tizi-Ouzou

Faculté de : Génie électrique et d'informatique
Département : Informatique



Mémoire de fin d'études

En vue de l'obtention du diplôme de

Master en Informatique

Spécialité : systèmes informatiques

Réalisé par :

KABECHE Samir et **AZEB Moussa**

Thème : l'apprentissage profond pour la construction du profil utilisateur en recherche d'information personnalisée

Proposé et dirigé par : Mme F. ACHEMOUKH

Soutenu le 04 Novembre 2020 devant le jury composé de :

Présidente : Mme FELLAG Samia

Examinatrice : Mme BENTAYEB Mouna

Promotrice : Mme ACHEMOUKH Farida

Résumé

La Recherche d'Information (RI) est une discipline dont le but est de faciliter l'accès à l'information pertinente à travers l'exploitation de modèles et de techniques d'acquisitions, d'organisations et de recherche de données.

Toutefois, lors de la phase d'appariement, la différence vocabulaire utilisée pour l'expression des contenus des documents et des besoins en information n'est pas prise en compte.

Afin de résoudre ce problème, l'intégration de l'utilisateur dans l'ensemble des phases de recherche a été prise en compte afin de lui délivrer l'information pertinente adaptée à son contexte et ses préférences répondant à ses besoins précis.

Notre contribution présentée dans ce mémoire porte sur la construction d'un profil utilisateur en se basant sur les réseaux de neurones.

Mots clés : recherche d'information, recherche d'information personnalisée, profil utilisateur, centre d'intérêt, réseaux de neurones.

Remerciements

Nous tenons tout d'abord à remercier Dieu le tout puissant et miséricordieux, qui nous a donné la force et la patience d'accomplir ce modeste travail.

En second lieu, nous tenons à remercier notre promotrice madame ACHEMOUKH Farida, qui nous a offert l'opportunité de réaliser ce mémoire et pour sa patience, sa disponibilité et surtout ses judicieux conseils qui ont contribué à alimenter nos réflexions.

Nous tenons à remercier également les membres de jury d'avoir accepté d'évaluer notre travail.

Nous tenons à remercier spécialement tous les enseignants, qui nous ont accompagnés tout au long de notre cursus et pour la qualité de leur travail.

Enfin, nous voudrions exprimer notre reconnaissance envers nos familles et amis qui nous ont apportés leur support moral et intellectuel tout au long de notre cursus.

On vous dit merci.

Dédicaces

Du profond de mon cœur je dédie ce modeste travail à tous ceux qui me sont chers,

A mes chers parents qui sont la source de mon courage et de ma persévérance, qu'aucune dédicace ne saurait exprimer l'amour et la reconnaissance que j'ai toujours pour vous, je vous dédie aujourd'hui ma réussite. Puisse Dieu, le tout puissant, vous préserver et vous accorder santé, longue vie et bonheur.

,

A mes chers frères Loucif et Meziane,

A ma chère sœur Sonia,

A mon binôme Moussa,

A ma deuxième famille " ASAM JUDO ",

Ainsi qu'à mes proches et mes amis(es).

Samir

Je dédie ce modeste travail à tous ceux qui me sont chères,

*Tout d'abord je souhaite dédier ce travail à la mémoire de mon père que dieu
l'accueil dans son vaste paradis.*

*Je dédie aussi ce travail à ma chère maman source de tendresse et d'amour qui
m'a tout donné et pour les sacrifices qu'elle a consentis mon instruction et son
soutien permanent au long de mes années d'étude, que dieu la garde et l'entoure
de sa bénédiction.*

*A mes chères soeurs " Hayet, Kahina et Nabila" qui de par leur soutien au
quotidien ont contribués à la réalisation de ce travail,*

A mon cher ami Amayas pour son soutien au long de la réalisation de ce travail,

*A mes chers amis, camarade et mon binôme Samir et a tous ceux qui ont
contribué de près ou de loin à la réalisation de ce travail.*

Moussa

Table des matières

INTRODUCTION GENERALE.....	1
-----------------------------------	----------

CHAPITRE 1 : CONCEPTS DE BASE DE LA RECHERCHE D'INFORMATION CLASSIQUE ET EMERGENCE DE LA RI PERSONNALISEE.	3
--	----------

I.1	INTRODUCTION.....	4
I.2	LES FONDEMENTS DE LA RECHERCHE D'INFORMATION	4
I.2.1	Notions de base	5
I.3	LE PROCESSUS DE RECHERCHE D'INFORMATION	6
I.3.1	L'indexation	6
I.3.2	L'appariement document-requête.....	9
I.4	LES MODELES DE RECHERCHE D'INFORMATION	10
I.4.1	Le modèle ensembliste.....	11
I.4.2	Le modèle algébrique ou vectoriel	12
I.4.3	Le modèle probabiliste.....	14
I.5	MESURES D'EVALUATION.....	15
I.6	DE LA RI CLASSIQUE A LA RI ADAPTATIVE	19
I.7	LA RI ADAPTATIVE.....	19
I.7.1	Reformulation de requête.....	20
I.7.2	Adaptation du contenu documentaire	21
I.8	FACTEURS D'EMERGENCE DE LA RI PERSONNALISEE	21
I.9	CONCLUSION.....	23

CHAPITRE 2 : L'ACCES PERSONNALISE A L'INFORMATION ET MODELISATION DU PROFIL UTILISATEUR.....	24
---	-----------

II.1	INTRODUCTION.....	25
II.2	LA MODELISATION DES UTILISATEURS	25
II.2.1	Modèles de représentation de profils utilisateurs.....	26

II.2.1.1	Modèle vectorielle.....	26
II.2.1.2	Modèle connexionniste.....	27
II.2.1.3	Modèle conceptuelle.....	28
II.2.1.4	Modèle multidimensionnel.....	29
II.2.2	Construction du modèle de l'utilisateur.....	31
II.2.2.1	Acquisition de données utilisateur.....	31
II.2.2.2	Techniques de construction du profil de l'utilisateur.....	33
II.2.3	Évolution du profil utilisateur.....	35
II.3	LES APPROCHES DE PERSONNALISATION.....	36
II.3.1	L'enrichissement des requêtes.....	36
II.3.2	Le filtrage des résultats.....	36
II.3.3	Reclassement de résultats.....	37
II.4	CONCLUSION.....	38

CHAPITRE 03 : APPRENTISSAGE PROFOND “ DEEP LEARNING”.....40

III.1	INTRODUCTION.....	41
III.2	DEFINITIONS DE L'APPRENTISSAGE AUTOMATIQUE.....	41
III.2.1	Types d'apprentissage automatique.....	42
III.2.1.1	L'apprentissage supervisé.....	42
III.2.1.2	L'apprentissage non supervisé.....	43
III.2.1.3	Apprentissage par Renforcement.....	44
III.2.1.4	Apprentissage semi-supervisé.....	44
III.3	L'APPRENTISSAGE PROFOND.....	44
III.3.1	Définition.....	44
III.4	LES RESEAUX DE NEURONES.....	46
III.4.1	Neurone formel.....	46
III.4.1.1	Définition d'un réseau de neurones artificiels.....	48
III.4.2	Fonction d'activation.....	49
III.4.2.1	La fonction Sigmoidale.....	49
III.4.2.2	La fonction Tangente Hyperbolique (Tanh).....	50
III.4.2.3	Fonction Relu.....	51
III.4.2.4	Fonction SoftMax.....	51
III.4.3	Fonctionnement des réseaux de neurones.....	52
III.4.3.1	La propagation direct.....	53

III.4.3.2	La rétro-propagation.....	53
III.4.4	Architecture de réseaux de neurones	54
III.4.4.1	Réseaux de neurones non bouclés (réseaux de neurones artificiel)	55
III.4.4.2	Réseaux récurrents	57
III.4.4.3	Réseaux convolutionnels	60
III.4.5	Algorithmes d'apprentissage des réseaux de neurones	63
III.4.5.1	Rétro propagation du gradient.....	63
III.4.5.2	Adagrad	64
III.4.5.3	Adam	65
III.4.5.4	Nastrov-Adam.....	65
III.5	APPRENTISSAGE AUTOMATIQUE ET CLASSIFICATION DE TEXTES	65
III.5.1	L'acquisition de textes brutes	66
III.5.2	Prétraitement de textes	66
III.5.2.1	Nettoyage de mots	67
III.5.2.2	Suppression des mots vides.....	67
III.5.2.3	La tokenisation.....	67
III.5.3	Représentation de textes.....	67
III.5.3.1	Représentation en sac de mots (Bag of Words).....	68
III.5.3.2	Représentation avec les n-grammes.....	68
III.5.3.3	Représentation par prolongement de mots (Word Embedding).....	69
III.5.4	Choix de classifieur	69
III.5.4.1	Les classifieurs naïf bayes.....	69
III.5.4.2	Les machines à vecteurs de support.....	70
III.5.4.3	Les classifieurs à base de réseaux de neurones	70
III.5.5	Corpus de Textes	70
III.5.6	Difficultés particulière de la classification de textes.....	71
III.5.6.1	Polysémie.....	71
III.5.6.2	Les mots composés.....	71
III.5.6.3	Redondance.....	72
III.5.6.4	Subjectivité de la décision.....	72
III.6	CONCLUSION	72

**CHAPITRE 04 : DEFINITION D'UN PROFIL UTILISATEUR EN SE
BASANT SUR LES RESEAUX DE NEURONES.73**

IV.1	INTRODUCTION.....	74
IV.2	MOTIVATIONS ET OBJECTIFS	74

IV.3	ARCHITECTURES DES MODELES PROPOSES	75
IV.3.1	Architecture des modèles de la première phase.....	75
IV.3.1.1	Premier modèle	75
IV.3.1.2	Deuxième modèle.....	77
IV.3.1.3	Troisième modèle.....	78
IV.3.2	Architectures des modèles de la deuxième phase	79
IV.3.2.1	Premier modèle	79
IV.3.2.2	Deuxième modèle.....	80
IV.4	CONCLUSION	81

CHAPITRE 5 : IMPLEMENTATION ET EXPERIMENTATIONS.....82

V.1	INTRODUCTION.....	83
V.2	PRESENTATION DES OUTILS	83
V.2.1	Logiciels.....	83
V.2.2	Bibliothèques logicielles	84
V.2.3	Configuration matérielle utilisée	86
V.3	JEUX DE DONNEES UTILISES	87
V.3.1	Première collection	87
V.3.2	La deuxième collection	88
V.4	MESURES D’EVALUATION	89
V.5	PRETRAITEMENT ET REPRESENTATION	90
V.5.1	Première collection	90
V.5.2	Deuxième Collection	91
V.6	ENTRAINEMENT ET RESULTATS.....	93
V.6.1	Première phase de l’approche.....	93
V.6.1.1	Résultats obtenus pour le premier modèle (ANN).....	93
V.6.1.2	Résultats obtenus pour le deuxième modèle.....	95
V.6.1.3	Une variante du deuxième modèle.....	97
V.6.1.4	Résultats obtenus pour le troisième modèle	99
V.6.1.5	Discussion et comparaison des résultats	101
V.6.2	Deuxième phase de l’approche.....	102
V.6.2.1	Résultats obtenus pour le premier modèle.....	102
V.6.2.2	Résultats obtenus pour le deuxième modèle.....	104

V.6.2.3	Discussion et comparaison des résultats	106
V.7	CONCLUSION	106
	CONCLUSION GENERALE	108
	BIBLIOGRAPHIE	109

Liste des figures

Figure I.1 le processus de la recherche d'information	6
Figure I.2 Modèles de RI selon (Baeza-Yates and Ribeiro-Neto, 1999).....	11
Figure I.3 Représentation des documents lors d'une interrogation du système de RI.....	16
Figure II.1 Exemple de profil représenté par des mots clés.....	27
Figure II.2 les phases de construction du profil utilisateur.....	31
Figure II.3 Représentation d'un système de filtrage	37
Figure III.1 La relation entre les trois branches l'intelligence artificielle(IA), (ML) et (DL). 45	
Figure III.2 Equivalence entre un neurone biologique et un neurone formel.....	46
Figure III.3 Représente la structure générale d'un neurone formel.	47
Figure III.4 La fonction sigmoïde	50
Figure III.5 La fonction Tangente Hyperbolique.	50
Figure III.6 La fonction relu.	51
Figure III.7 : La fonction SoftMax.....	52
Figure III.8 la représentation de schéma d'un perceptron.	56
Figure III.9 Réseau de neurones à couches entièrement connectées.....	57
Figure III.10 Architecture d'un réseau récurrent.	58
Figure III.11 Architecture d'un bloc LSTM.	59
Figure III.12 Architecture de Réseaux GRU	59
Figure III.13 Architecture d'un Réseaux Convolutionnels.....	61
Figure III.14 Processus de classification de textes.....	66
Figure IV.1 Architecture du premier modèle.....	76
Figure IV.2 Architecture du deuxième modèle.....	77
Figure IV.3 Architecture du troisième modèle	78
Figure IV.4 Architecture du premier modèle de la deuxième phase.....	80
Figure IV.5 Architecture du deuxième modèle de la deuxième phase.....	81
Figure V.1 Structure de la première collection	87
Figure V.2 Structure de la deuxième collection.....	88
Figure V.3 Prétraitement et représentation de la première collection	91
Figure V.4 Prétraitement et représentation de la deuxième collection.....	92
Figure V.5 Prétraitement et représentation de la deuxième collection.....	93

Figure V.6 precision (accuracy) du premier modèle	94
Figure V.7 erreur (loss) du premier modèle.....	94
Figure V.8 Configuration du deuxième modèle.....	95
Figure V.9 précision (accuracy) du deuxième modèle	96
Figure V.10 erreur (loss) du deuxième modèle.....	96
Figure V.11 Configuration de la variante du deuxième modèle	97
Figure V.12 précision(accuracy) de la variante du deuxième modèle	98
Figure V.13 erreur (loss) de la variante du deuxième modèle	98
Figure V.14 Configuration de la variante du troisième modèle.....	99
Figure V.15 précision (accuracy) du troisième modèle.....	100
Figure V.16 erreur(loss) du troisième modèle	100
Figure V.17 Configuration du premier modèle.....	102
Figure V.18 précision (accuracy) du premier modèle	103
Figure V.19 erreur(loss) du premier modèle.....	103
Figure V.20 Configuration du deuxième modèle.....	104
Figure V.21 précision (accuracy) du deuxième modèle	105
Figure V.22 erreur (loss) du deuxième modèle.....	105

Liste des tableaux

Tableau V.1 résultats obtenus pour le premier modèle	95
Tableau V.2 résultats obtenus pour le deuxième modèle	97
Tableau V.3 résultats obtenus pour la variante du deuxième modèle	99
Tableau V.4 résultats obtenus pour la variante du troisième modèle.....	100
Tableau V.5 Comparaison des résultats obtenus pour les trois modèles de la première collection.....	101
Tableau V.6 résultats obtenus pour le premier modèle.	103
Tableau V.7 résultats obtenus pour le deuxième modèle.	105
Tableau V.8 Comparaison des résultats obtenue des deux modèles avec la deuxième collection.....	106

Introduction Générale

La recherche d'information (RI), est une branche en informatique qui s'intéresse à l'acquisition, l'organisation, le stockage et la recherche des informations. Elle consiste en un ensemble de procédures et techniques permettant de sélectionner, les informations (documents ou parties de documents) pertinentes en réponse à un besoin en information exprimé par l'utilisateur à travers une requête.

Toutefois, lors de la phase d'appariement, la différence vocabulaire utilisé pour l'expression des contenus des documents et des besoins en information n'est pas prise en compte.

Ce défaut d'appariement engendre la dégradation des performances de recherche. Ainsi le problème n'est plus tant la disponibilité de l'information mais la capacité de sélectionner l'information répondant aux besoins d'un utilisateur.

Afin de résoudre ce problème, l'intégration de l'utilisateur dans le processus de recherche a été prise en compte afin de lui délivrer l'information pertinente adaptée à son contexte et ses préférences répondant à ses besoins précis.

Dans un souci d'être au plus près de ces besoins, il est important de les connaître au mieux, en précisant le profil utilisateur.

La construction de ce profil consiste alors en la collecte des informations qui le représentent et cela d'une manière explicite, en se basant sur les informations fournies directement par l'utilisateur, ou implicitement, à partir des documents consultés et des comportements de l'utilisateur.

Notre objectif est de définir un profil utilisateur en se basant sur les réseaux de neurones. Ce type de réseaux permet la prédiction du profil utilisateur en se basant sur des documents jugés pertinents par l'utilisateur.

Dans notre travail, le profil de l'utilisateur est défini par un ensemble de centres d'intérêts. Il est construit et inféré d'une manière implicite et automatique à base des réseaux de neurones. Le profil résultant est organisé selon des structures ensemblistes.

Organisation du mémoire

Ce mémoire est organisé en cinq chapitres :

Chapitre 1 : Concepts de base de la Recherche d'Information classique et émergence de la RI personnalisée.

Chapitre 2 : L'accès personnalisé à l'information et modélisation du profil utilisateur.

Chapitre 3 : Apprentissage profond 'deep learning'.

Chapitre 4 : Modèles de réseaux de neurones proposés pour définir le profil utilisateur.

Chapitre 5 : Implémentation et expérimentations.

**Chapitre 1 : Concepts de base de la Recherche
d'Information classique et émergence de la RI
personnalisée.**

I.1 Introduction

La Recherche d'Information (RI) est une discipline dont le but est de faciliter l'accès à l'information pertinente à travers l'exploitation de modèle et de techniques d'acquisition, d'organisation et de recherche de données.

Les premiers modèles proposés en recherche d'information quantifient la pertinence d'un document seulement en fonction de la requête. Or pour une même requête soumise par deux utilisateurs distincts et présentant des besoins différents, ces modèles retournent des résultats similaires.

De ce fait, les performances d'un SRI ne sont plus dépendantes de l'indexation des documents et de l'appariement document-requête uniquement, mais aussi sa capacité à modéliser l'utilisateur et son exploitation dans le processus de recherche d'information.

Dans ce chapitre, nous présentons tout d'abord les concepts de base liés à la recherche d'information, en particulier les notions de document, de requête et de pertinence. la section 1.3 traite le processus de la recherche d'informations et ses différentes étapes, à savoir l'indexation, l'appariement et la reformulation d'un besoin en information. La section 1.4 passe en revue les principaux modèles de RI. La section 1.5 traite les mesure d'évaluation. La section 1.6 présente les principales causes d'émergence de la RI adaptative. Dans la section 1.7 nous introduisons la RI adaptative. La section 1.8 présente les principales causes d'émergence de la RI personnalisée. La dernière section conclut le chapitre.

I.2 Les fondements de la recherche d'information

L'objectif principal de la recherche d'information est de fournir des techniques et des outils pour sélectionner les informations pertinentes contenues dans une collection de documents (un corpus) en réponse aux besoins en information d'un utilisateur représentés à l'aide d'une requête.

Nous citons ci-dessous la définition de la RI donnée par [Van Rijsbergen, 1986] dans sa forme originelle : "The user expresses his information need in the form of a request for information.

Chapitre 1 : Concepts de base de la Recherche d'Information classique et émergence de la RI personnalisée.

Information retrieval is concerned with retrieving those documents that are likely to be relevant to his information need as expressed by his request".

Dans la définition de la RI nous avons cité trois notions de base qui sont : documents, requête et la pertinence. Nous allons les définir dans ce qui suit.

I.2.1 Notions de base

Document : On appelle document toute unité d'information qui peut constituer une réponse à un besoin en information/requête d'un utilisateur. Un document peut être un texte, un morceau de texte, une image, une bande vidéo, etc.

Requête : Une requête constitue l'expression du besoin en information de l'utilisateur. Elle représente l'interface entre le SRI et l'utilisateur. Elle est souvent représentée par un ensemble de mots clefs exprimés en langage naturel, booléen ou par des sélections graphiques [**Lelu and Francois, 1992**]

La pertinence : C'est la correspondance entre un document et un besoin d'information, et un besoin en information d'un utilisateur exprimé généralement par une requête. On distingue de type de pertinence, la pertinence utilisateur et système. [**Cosijn and Ingwersen, 2000**]

- ❖ **La pertinence système :** est souvent traduite par un score cherchant à évaluer la pertinence des documents vis-à-vis une requête. Ce type de pertinence est souvent déterministe.
- ❖ **La pertinence utilisateur :** Ce type de pertinence est lié au jugement apporté par l'utilisateur à un document retourné par le système. Donc deux utilisateurs peuvent donner deux jugements différents sur un document retourné. Ce qui fait que cette pertinence est subjective.

I.3 Le processus de recherche d'information

Le but fondamental d'un SRI est de sélectionner l'ensemble de documents pertinents répondant au besoin en information de l'utilisateur, d'autres termes son rôle est de mettre en œuvre des moyens, des techniques afin de retrouver des documents pertinents dans un corpus pour répondre aux requêtes émises par des utilisateurs. La réalisation d'un tel système consiste principalement à mettre en œuvre un processus clé illustré dans la figure I.1 .

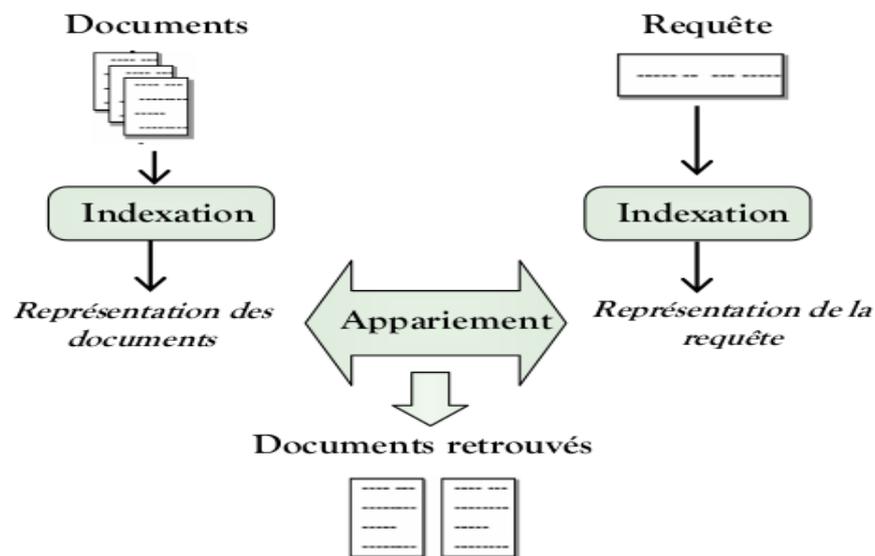


Figure I.1 le processus de la recherche d'information

Ce processus consiste en deux principales phases, à savoir, l'indexation et l'appariement

I.3.1 L'indexation

L'indexation consiste à analyser les documents que le système utilisera afin d'en extraire les mots clefs, appelés aussi descripteurs, qui serviront à la recherche. Le résultat de l'indexation est souvent représenté comme une liste de termes ou groupe de termes significatifs pour l'unité

Chapitre 1 : Concepts de base de la Recherche d'Information classique et émergence de la RI personnalisée.

textuelle correspondante, généralement assortis de poids représentant leur degré de représentativité du contenu sémantique de l'unité qu'ils décrivent. Les descripteurs des documents (mots, groupe de mots) sont rangés dans un catalogue appelé dictionnaire constituant le langage d'indexation.

Techniquement, l'indexation peut être manuelle, automatique ou semi-automatique : [Salton et al., 1986] :

- **Manuelle** : chaque document est analysé par un spécialiste du domaine ou un documentaliste.

- **Automatique** : chaque document est analysé à l'aide d'un processus entièrement automatisé.

- **Semi-automatique (mixte)** : c'est une combinaison des deux méthodes précédentes : un premier processus automatique permet d'extraire les termes du document. Cependant, le choix final reste au spécialiste du domaine ou au documentaliste pour établir les relations entre les mots clés et choisir les termes significatifs.

Les termes extraits des documents ne jouent pas le même rôle dans la représentation de ces derniers, Pour caractériser le degré de discrimination de ces termes. Pour caractériser ce degré de discrimination, il est courant en RI, d'affecter à chaque terme un poids. Cette étape est primordiale dans le processus d'indexation correspond au processus de pondération. Pour trouver les termes du document qui représentent le mieux son contenu sémantique, la formule de pondération la plus utilisée est celle basée sur la fréquence des termes dans les documents, appelée TF-IDF. [Robertson and Sparck Jones, 1976]

Chapitre 1 : Concepts de base de la Recherche d'Information classique et émergence de la RI personnalisée.

- TF (term frequency) : cette mesure est proportionnelle à la fréquence du terme dans le document. L'idée sous-jacente est que plus un terme est fréquent dans un document, plus il est important dans la description de ce document, souvent exprimé selon l'expression suivante :

$$\log(1 + T F).$$

- IDF (Inverse Document Frequency) : mesure l'importance d'un terme dans toute la collection. L'idée sous-jacente est que les termes qui apparaissent dans peu de document de la collection sont plus représentatifs du contenu de ces documents que ceux qui apparaissent dans tous les documents de la collection.

Cette mesure est exprimée selon l'une des déclinaisons suivantes :

$$1. IDF = \log \left(\frac{N}{df} \right)$$

$$2. IDF = \log \left(\frac{N-df}{df} \right)$$

Où df est la proportion de documents contenant le terme et N le nombre total de documents dans la collection.

La fonction de pondération de la forme $TF * IDF$ consiste à multiplier les deux mesures TF et IDF. Une formule largement utilisée est la suivante :

$$T F * IDF = \log (1 + T F) * \log \left(\frac{N}{df} \right) \quad (1.1)$$

Chapitre 1 : Concepts de base de la Recherche d'Information classique et émergence de la RI personnalisée.

Une normalisation de la mesure du TF * IDF par rapport à la longueur des documents a été proposée par ([Singhal,1997], [Robertson et al., 1995]). Une des formules les plus utilisées dans le domaine de la RI est la formule BM25 d'OKAPI [Robertson et al., 1999] tel que le poids d'un terme t_i dans le document d_j (noté $w(i, j)$) est donnée par :

$$w(i, j) = 0.5 * \frac{tf_{ij} * \log\left(\frac{N - n_i + 0.5}{n_i + 0.5}\right)}{2 * \left(0.25 + \frac{0.75 + dl_j}{avg - dl}\right) + tf_{ij}} \quad (1.2)$$

Où :

n_i : Le nombre de documents contenant le terme t_i .

N : Le nombre de documents pertinents dans la collection.

dl : La longueur du document d_j .

$avg - dl$: La longueur moyenne des documents de la collection.

tf_{ij} : La fréquence d'apparition du terme t_i dans le document d_j .

I.3.2 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 (similarité vectorielle, probabiliste, etc.). Ce score de pertinence est calculé à partir d'une fonction ou d'une mesure de similitude, notée RSV (Q, D) (Retrieval Status Value) où Q est une requête et D un document de la collection. Le processus d'appariement est étroitement lié au processus d'indexation et de pondération des termes. Il existe deux méthodes d'appariement :

Chapitre 1 : Concepts de base de la Recherche d'Information classique et émergence de la RI personnalisée.

➤ **Appariement exact** (« exact match retrieval »)

Le résultat est une liste de documents respectant exactement la requête spécifiée avec des critères précis. Les documents retournés ne sont pas triés [Salton, 1971].

➤ **Appariement approché** (« best match retrieval »)

Le résultat est une liste de documents censés être pertinents pour la requête. Les documents retournés sont triés selon leur score de pertinence vis-à-vis de la requête [Robertson and Sparck Jones, 1976].

I.4 Les modèles de recherche d'information

Le modèle de recherche d'information joue un rôle principal dans la RI : il détermine le comportement du système de RI. Il se définit principalement par sa manière de modéliser la fonction de correspondance document/requête ainsi que par la représentation des documents et des requêtes. On peut distinguer trois grandes classes de modèles [Baeza-Yates and Ribeiro-Neto, 1999].

Les modèles ensemblistes : sont basés sur la théorie des ensembles et ont été les premiers à avoir été mis en place.

Les modèles algébriques : calculent des distances entre les représentations.

Les modèles probabilistes : sont basés sur la théorie des probabilités. Ils ont pour objectif d'estimer des probabilités de pertinence d'un document en fonction de la requête.

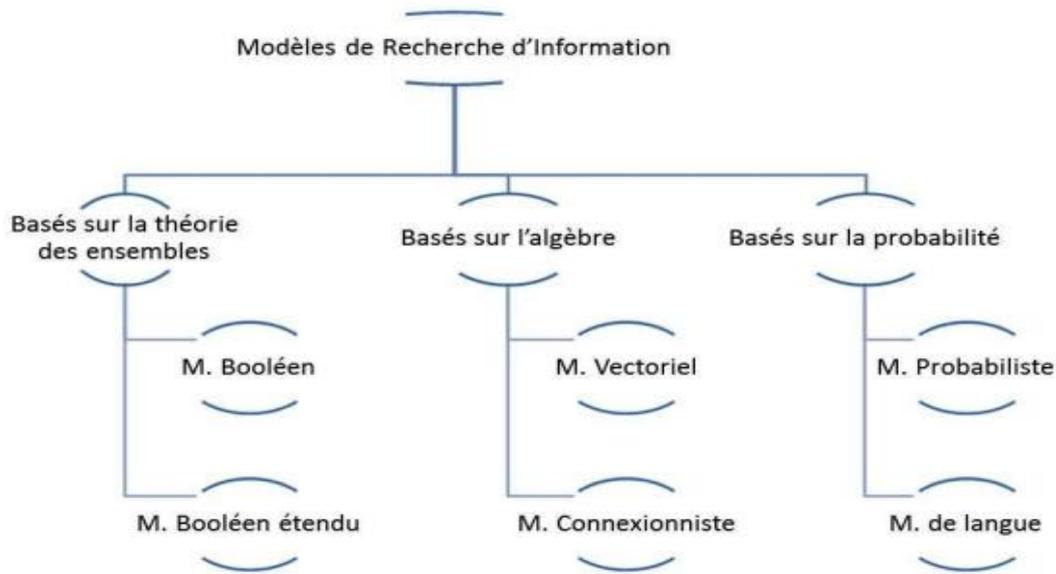


Figure I.2 Modèles de RI selon (Baeza-Yates and Ribeiro-Neto, 1999).

Nous présentons dans la suite de cette section ces trois modèles.

I.4.1 Le modèle ensembliste

Les modèles ensemblistes sont basés sur la théorie des ensembles proposés par Salton [Salton, 1971].

Dans un modèle de ce type, documents et requêtes sont représentés par des ensembles de mots clés. Les documents sont indexés par un ensemble de termes non pondérés. Ils sont de ce fait représentés par une conjonction logique de ces termes. Quant aux requêtes, elles sont sous forme d'expressions booléennes dont les termes sont reliés par des opérateurs logiques OR, AND, NOT. Pour un document $d = t_1, t_2, \dots, t_n$, et une requête q , la fonction de correspondance $rsv(d, q)$, pour Retrieval status value, est déterminée comme suit :

Chapitre 1 : Concepts de base de la Recherche d'Information classique et émergence de la RI personnalisée.

$$rsv(d, ti) = 1 \text{ si } ti \in d ; 0 \text{ sinon}$$

$$rsv(d, q1 \wedge q2) = 1 \text{ si } rsv(d, q1) = 1 \text{ et } rsv(d, q2) = 1 ; 0 \text{ sinon.}$$

$$rsv(d, q1 \vee q2) = 1 \text{ si } rsv(d, q1) = 1 \text{ ou } rsv(d, q2) = 1 ; 0 \text{ sinon.}$$

$$rsv(d, \neg q1) = 1 \text{ si } rsv(d, q1) = 0 ; 0 \text{ sinon.}$$

En effet, les documents à retrouver sont exactement décrits par la requête : ils contiennent tous les termes de la requête émise (intersection des ensembles). Les documents, de ce fait, sont de deux types : pertinents ou non pertinents.

Ce modèle a l'avantage d'être facile à implémenter et de permettre aux utilisateurs d'exprimer des contraintes structurelles et conceptuelles. Néanmoins, ce modèle présente les problèmes suivants :

Les opérateurs logiques peuvent être mal manipulés par les usagers ce qui rend difficile la formulation de requêtes adéquates à l'aide d'expressions booléennes. En effet, dans le langage courant l'expression de recherche "recherche d'information et personnalisation" veut dire recherche les deux ensembles de termes.

Or, en logique booléenne, le "et" représente l'intersection de ces deux ensembles de termes.

La correspondance entre document et requête étant soit 1, soit 0, il n'est pas possible à un système de recherche d'information de classer les documents retournés par ordre de pertinence.

I.4.2 Le modèle algébrique ou vectoriel

Dans le modèle vectoriel, les requêtes et le contenu des documents sont représentés dans un espace vectoriel construit par les termes d'indexation. La requête ainsi que le document sont donc représentés par deux vecteurs respectifs. La fonction de correspondance est alors calculée

Chapitre 1 : Concepts de base de la Recherche d'Information classique et émergence de la RI personnalisée.

en se basant sur la similarité entre ces deux vecteurs. Pour cela, des mesures comme le produit scalaire de deux vecteurs, la distance métrique et la mesure cosinus ont été proposées. Soient R l'espace vectoriel défini par l'ensemble des termes t_1, t_2, \dots, t_N , le document d et la requête q peuvent être représentés par des vecteurs de poids tels que :

$\vec{d} = (w_1, w_2, \dots, w_N)$; w_i correspond au poids associé au terme d'indice i de R dans le document d et $Q = (q_1, q_2, \dots, q_N)$; q_i correspond à un coefficient. Nous présentons quelques mesures de similarité des vecteurs.

- Le produit scalaire :

$$rsv(\vec{d}, \vec{q}) = \sum_{k=1}^n (d_k * q_k) \quad (1.3)$$

- La mesure de Jaccard :

$$Jaccard(D_j * Q) = \frac{\sum_{i=1}^n (d_{ij} * q_i)}{\sum_{i=1}^n (q_i^2) * \sum_{i=1}^n (d_{ij}^2) - \sum_{i=1}^n (d_{ij} * q_i)} \quad (1.4)$$

- La mesure cosinus :

$$\cos(D_j * Q) = \frac{\sum_{i=1}^n d_{ij} * q_i}{\sqrt{\sum_{i=1}^n q_i^2 * \sum_{i=1}^n d_{ij}^2}} \quad (1.5)$$

Chapitre 1 : Concepts de base de la Recherche d'Information classique et émergence de la RI personnalisée.

Pour la mesure de cosinus, une représentation vectorielle complète est utilisée avec la fréquence des mots. Dans ce cas, deux documents sont similaires si leurs vecteurs sont confondus. Et le cas échéant, leurs vecteurs forment un angle dont le cosinus est représenté par l'équation 1.6.

I.4.3 Le modèle probabiliste

Les modèles probabilistes se basent sur un calcul de similarité probabiliste. En effet, ce modèle est fondé sur le calcul de la probabilité de pertinence d'un document pour une requête et son principe de base est de retrouver, pour une requête donnée, les documents ayant à la fois une forte probabilité d'être pertinents, et une faible probabilité de ne pas l'être. La similarité entre une requête q et un document d est donc en fonction de la probabilité de pertinence du document d pour la requête q : il est soit pertinent soit non pertinent pour la requête. De ce fait, un document est sélectionné si la probabilité de sa pertinence est supérieure à la probabilité de sa non pertinence.

En notant R l'évènement de pertinence de d pour q et \bar{R} l'évènement de non pertinence, la fonction de correspondance document requête, appelée aussi pertinence globale, $RSV(Q, D)$, est calculée par :

$$RSV(q, d) = \frac{p(R/q)}{P(\bar{R}/d)} \quad (1.6)$$

Où $P(R/q)$ est la probabilité que le document soit pertinent, et $P(\bar{R}/d)$ la probabilité que le document soit non pertinent.

Le modèle probabiliste a vu plusieurs extensions, la plus connue est BM25. Dans le modèle Okapi BM25, le calcul du poids d'un terme dans un document prend en considération la fréquence des termes, leur rareté ainsi que la longueur des documents. Le poids d'un terme dans un document est donc calculé par la fonction suivante :

$$w = \log \frac{\frac{(r+0.5)}{(R-r+0.5)}}{\frac{(n-r+0.5)}{(N-n-R+r+0.5)}} \quad (1.7)$$

Avec :

r : Nombre de documents pertinents contenant le terme t .

R : Nombre total de documents pertinents.

n : Nombre de documents contenant le terme t .

N : Nombre total de documents.

1.5 Mesures d'évaluation

Les mesures d'évaluation se sont enrichies au fil des ans, la plupart se basent sur l'hypothèse que les documents non jugés sont non pertinents. A cet effet, plusieurs métriques ont été proposées pour évaluer la performance d'un SRI. La majorité d'entre elles tentent d'évaluer la capacité d'un SRI à sélectionner les documents pertinents en réponse à une requête. Nous présentons dans ce qui suit quelques-unes des mesures les plus utilisées dans l'évaluation de systèmes de RI.

Soit D l'ensemble des documents présents dans le corpus de recherche. Pour une requête donnée, un document est soit restitué ou non restitué et soit pertinent ou non pertinent.

La figure 1.3 représente la répartition des documents lors d'une interrogation d'un système de RI.

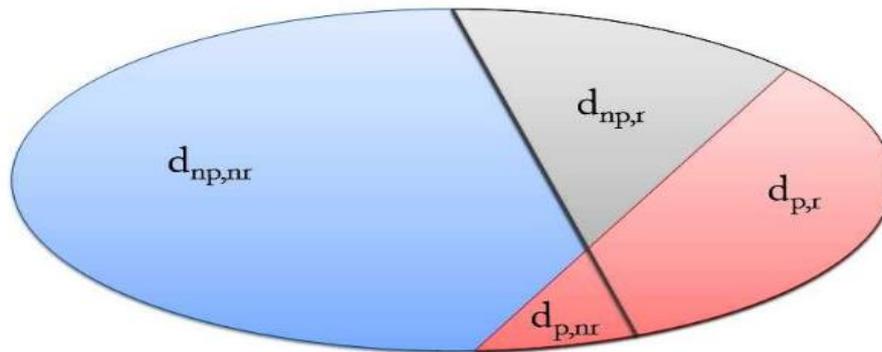


Figure I.3 Représentation des documents lors d'une interrogation du système de RI

Avec :

$d_{np,nr}$: L'ensemble des documents non pertinents et non restitués lors de la recherche.

$d_{np,r}$: L'ensemble des documents non pertinents et restitués, il s'agit du bruit.

$d_{p,r}$: L'ensemble des documents pertinents et restitués à l'utilisateur.

$d_{p,nr}$: L'ensemble des documents pertinents et non restitués, il s'agit du silence.

Nous reprenons les mêmes notations de la figure 1.3 afin de présenter les mesures d'évaluation suivantes :

- **La précision** : mesure la capacité du système à rejeter tous les documents non pertinents pour une requête. Elle est donnée par le rapport entre les documents retournés pertinents et l'ensemble des documents retournés. Elle est exprimée comme suit:

$$P = \frac{d_{p,r}}{d_{p,r} + d_{np,r}} \quad (1.8)$$

Chapitre 1 : Concepts de base de la Recherche d'Information classique et émergence de la RI personnalisée.

-Le rappel : mesure la capacité du système à renvoyer tous les documents pertinents pour une requête. Il est donné par le rapport entre les documents pertinents retournés et l'ensemble des documents pertinents pour la requête. Il est exprimé comme suit :

$$R = \frac{d_{p,r}}{d_{p,r} + d_{p,nr}} \quad (1.9)$$

Les mesures de rappel et de précision sont inversement proportionnelles : lorsque l'une augmente l'autre diminue. Intuitivement, si on augmente le rappel en retrouvant plus de documents pertinents, on diminue la précision en retrouvant aussi plus de documents non pertinents. Inversement, une plus grande précision risque de rejeter des documents pertinents diminuant ainsi le rappel.

- F-mesure : est une mesure qui combine la précision et le rappel, nommée F-mesure ou F-score. Elle est définie par :

$$F - \text{mesure} = \frac{(1+\beta^2)*P*R}{(\beta^2*P)+R} \quad (1.10)$$

Le paramètre B permet de pondérer la précision ou le rappel. Il est généralement égal à 1.

La valeur de F-mesure diminue quand l'un de ses paramètres est petit et elle augmente quand les deux paramètres sont proches tout en étant élevés. En pratique, plus grande est la valeur de la F-mesure, meilleur est la recherche.

Chapitre 1 : Concepts de base de la Recherche d'Information classique et émergence de la RI personnalisée.

-La précision à X : est la précision à différents niveaux de coupe. P_n , est le calcul de la précision après que n documents aient été retrouvés (pertinents ou non) ; n peut être dans l'ensemble $\{5, 10, 15, 20, 30, 100, 200, 500, 1000\}$.

-la précision moyenne : AP (Average Precision) est la précision moyenne calculée pour une requête. Il s'agit de la précision moyenne obtenue chaque fois qu'un document pertinent est retrouvé. La formule utilisée est :

$$AP = \frac{\sum_{k=1}^n (Pk * rel(k))}{d_{p,r}} \quad (1.11)$$

Où :

k le rang du document retourné.

n le nombre de documents retournés.

PK précision au niveau de coupe k .

$rel(k) = 1$ si le k^{me} document restitué est pertinent.

$rel(k) = 0$ si le k^{me} document restitué est non pertinent.

-La moyenne des précisions moyennes : plus utilisée sous la terminologie anglaise "Mean Average Précision", MAP . Il s'agit de la moyenne des précisions pour l'ensemble des requêtes considérées.

$$MAP = \frac{\sum_{q \in Q} AP_q}{|Q|} \quad (1.12)$$

Chapitre 1 : Concepts de base de la Recherche d'Information classique et émergence de la RI personnalisée.

Où : Q est l'ensemble des requêtes et $|Q|$ est le nombre de requêtes.

I.6 De la RI classique à la RI adaptative

La RI classique repose principalement sur l'efficacité et la qualité des mécanismes d'indexation et d'appariement. Lors de l'appariement requête/document, seuls les documents qui sont les plus proches sémantiquement du besoin de l'utilisateur sont sélectionnés.

Néanmoins, dans la pratique la majorité des requêtes exprimées par les utilisateurs sont courtes et ambiguës, ce qui donne des spécifications inachevées sur leur besoin en informations. En outre, cette liste de termes ne correspond pas forcément à ceux utilisés pour indexer les documents pertinents de la collection et manque souvent de termes significatifs pouvant exprimer effectivement le besoin en information de l'utilisateur.

La différence de vocabulaire entre les termes choisis par l'utilisateur pour formuler sa requête et les termes utilisés pour représenter les documents engendrent un défaut d'appariement, De ce fait, les performances d'un SRI, ne dépendent pas uniquement de l'efficacité et la qualité des mécanismes d'indexation et d'appariement, mais de façon non négligeable de la capacité du SRI de prendre en considération les besoins de l'utilisateur pour mieux répondre à leurs attentes. De ce constat est apparu un nouvel axe de recherche, celui de la RI adaptative.

I.7 La RI adaptative

Les travaux de la RI adaptative se sont particulièrement axés sur l'amélioration de l'efficacité du processus de recherche notamment lors de la phase d'exécution de la requête. Les techniques développées ont eu pour but de désambigüiser le sens des mots de la requête utilisateur afin de mieux cerner le but de sa recherche. Plus particulièrement, la RI adaptative s'articule autour de deux approches :

* Adaptation de la phase d'expression du besoin en reformulant la requête initiale de l'utilisateur.

Chapitre 1 : Concepts de base de la Recherche d'Information classique et émergence de la RI personnalisée.

* Adaptation du contenu informationnel du fond documentaire en identifiant des connexions appropriées entre les documents et les requêtes dans un domaine applicatif spécifique.

I.7.1 Reformulation de requête

La reformulation de requêtes est un processus ayant pour objectif de générer une nouvelle requête plus adéquate que celle initialement formulée par l'utilisateur en ajoutant de nouveaux termes et/ou supprimant des termes inutiles. Cette reformulation permet de coordonner le langage de recherche (utilisé par l'utilisateur dans sa requête) et le langage d'indexation des documents. On distingue principalement deux approches de reformulation de requêtes : une approche basée sur un processus automatique et une autre, basée sur un processus interactif.

- **La reformulation automatique de requête :** La reformulation automatique de requête ou expansion de requête est l'une des premières techniques ayant produit des améliorations notables dans ce domaine. L'idée de base est d'ajouter à la requête initiale des termes issus de ressources linguistiques existantes ou bien de ressources construites à partir des collections. Plus précisément, au niveau des ressources linguistiques, le but est d'utiliser un Vocabulaire contrôlé issu de ressources externes. Il s'agit principalement de chercher des associations inter-termes extraites à partir des ontologies linguistiques (tel que WordNet [G.Miller, 1995], [Brajnik et al., 1996]). Les thésaurus construits manuellement sont un moyen efficace pour l'expansion de requête.
- **La reformulation interactive de requête (réinjection de pertinence) :** A la différence de la reformulation automatique, l'approche interactive (ou par réinjection de pertinence et/ou non-pertinence) exploite uniquement un sous-ensemble de documents sélectionnés parmi les premiers résultats obtenus de l'exécution de la requête initiale. Son principe fondamental est d'utiliser cette requête pour amorcer la recherche, puis modifier celle-ci à partir des jugements de pertinence et/ou de non pertinence de l'utilisateur, soit pour répondre les termes de la requête initiale [Robertson et al., 2000].

I.7.2 Adaptation du contenu documentaire

Dans cette approche, l'objectif de la RI adaptative est de définir des modèles de recherche, dits connexionnistes, permettant de décrire les représentations associatives entre les termes, les requêtes et les documents. L'idée de base est que les requêtes similaires ont un ensemble similaire de documents pertinents et que les informations capitalisées sur les documents pertinents pour ces requêtes devraient servir à retrouver les documents pertinents pour une nouvelle requête.

Les principaux travaux dans ce domaine se sont orientés vers l'application des réseaux de neurones. La particularité du réseau de neurones est de représenter les relations et associations qui existent entre les termes (ex. synonymie, voisinage, etc.), entre les documents (ex. similitude, référence, etc.), et enfin entre les termes et les documents (exemple, fréquence, poids, etc.).

I.8 Facteurs d'émergence de la RI personnalisée

Malgré le gain de performance apporté par les techniques développées en RI adaptative, elles présentent toutefois des limitations. Une analyse fine des travaux dans ce domaine montre que ces performances dépendent de nombreux facteurs a priori non contrôlés par le processus de réécriture adaptative de la requête. Ces facteurs, principalement liés aux approches de reformulation de requête, qui sont ainsi problématiques, peuvent être catégorisés selon trois principales dimensions : l'utilisateur, l'information portée par la requête et/ou document et l'interaction entre l'utilisateur et le SRI. Nous discutons dans ce qui suit chacune de ces dimensions [Tamine and Calabretto, 2008].

- **La dimension utilisateur**

Lors d'une recherche, le contexte est généralement limité au besoin informationnel extrait des termes de la requête soumise. Des informations telles que la tâche courante de l'utilisateur, ses

Chapitre 1 : Concepts de base de la Recherche d'Information classique et émergence de la RI personnalisée.

précédentes recherches, sa situation géographique, ces centres d'intérêt ou autres informations utiles sont rarement utilisées lors de l'interprétation de la requête de l'utilisateur.

- **La dimension information**

- Le volume important d'information accessible engendre incontestablement une diversité importante du vocabulaire. Par conséquent, les algorithmes d'ordonnement des termes d'expansion de requêtes en fonction de leur corrélation au sujet de la requête, sont peu performants.
- Les documents du Web contiennent de nombreuses informations non directement liées au sujet du document telles que les liens de navigation, les informations ou images publicitaires etc. Ces informations, même extraites des documents les mieux classés à l'issue d'une recherche initiale, engendrent du bruit lors d'un processus de réécriture de requête.

- **La dimension interaction**

Les techniques de reformulation de la requête par réinjection de pertinence nécessitent une rétroaction explicite de la part de l'utilisateur. Par ailleurs, plusieurs études montrent que la majorité des utilisateurs se limitent à fournir la requête initiale et préfère des mécanismes d'amélioration qui fonctionnent sans demande explicite d'information complémentaire.

Cette analyse montre globalement que les stratégies de RI adaptative ne sont pas garantes sur l'uniformité de la qualité des résultats d'un SRI dans des conditions d'utilisation différentes.

Le développement de services d'accès délivrant l'information pertinente de manière personnelle en fonction des caractères spécifiques de l'utilisateur et adaptant les résultats de recherche en fonction des préférences et contexte de l'utilisateur devient une nécessité absolue.

Chapitre 1 : Concepts de base de la Recherche d'Information classique et émergence de la RI personnalisée.

C'est pourquoi, les travaux se sont orientés vers la modélisation de l'utilisateur et son intégration comme composante du modèle global de recherche d'information. Ces travaux s'inscrivent dans le cadre précis de la « personnalisation de l'information ».

I.9 Conclusion

Au cours de ce chapitre, nous avons présenté les principaux facteurs d'émergence de la personnalisation dans le domaine de RI, à travers l'évolution des SRI classiques vers les SRI adaptatifs. Nous avons présenté les principaux concepts de la RI, à travers l'architecture commune à tous les SRI, permettant l'appariement entre les requêtes formulées par des utilisateurs et les documents de la collection. Nous avons également présenté les différents modèles et stratégies utilisés lors de la mise en œuvre de ces concepts ainsi que les mesures d'évaluations.

Nous présentons dans le prochain chapitre les principaux concepts et techniques liés à la recherche d'information personnalisée.

Chapitre 2 : L'accès personnalisé à l'information et modélisation du profil utilisateur.

II.1 Introduction

Adapter, personnaliser un document ou une application pour un utilisateur particulier nécessite une description plus élaborée de l'utilisateur et de sa représentation en tant qu'objet à part entière du système. Cette représentation de l'utilisateur vise à procurer au système les moyens de procéder aux adaptations souhaitées, d'évaluer la pertinence des objets disponibles (documents, pages Web, etc.) ou d'aider le système à faire des choix.

En amont de la représentation effective d'un utilisateur dans le système, et de l'exploitation de cette représentation, il est primordial de définir comment représenter un modèle utilisateur. En d'autres termes, il faut choisir les informations à retenir pour en construire un modèle. Plusieurs approches sont proposées pour guider l'élaboration de tels modèles et permettre de décrire une classification de ceux-ci.

Ce chapitre est consacré à la modélisation de l'utilisateur pour l'accès personnalisé à l'information. Nous y présentons les principales phases du processus de modélisation : les approches de représentation, les approches de construction, les méthodes d'évolution des profils utilisateurs et les approches de personnalisation.

II.2 La modélisation des utilisateurs

Le modèle utilisateur est défini par Mc Tear comme « une source de connaissances, une base de données sur un utilisateur » [Mctear, 1993]. Plus précisément, il s'agit d'un ensemble de données persistantes qui caractérisent un utilisateur ou un groupe d'utilisateurs particuliers. Un tel modèle peut contenir des caractéristiques sur les connaissances, les préférences, les objectifs, les centres d'intérêts, etc. d'un utilisateur.

Modéliser l'utilisateur, ses centres d'intérêts et ses préférences est une tâche primordiale pour les concepteurs des systèmes de RI personnalisée. Construire un modèle de l'utilisateur nécessite la connaissance de la structure à mettre en place afin de stocker les informations le concernant et de les exploiter d'une manière optimale. Plusieurs approches et techniques ont été développées afin de modéliser l'utilisateur. Ces techniques diffèrent par leur représentation, construction et mise à jour du contenu du modèle de l'utilisateur. Ce dernier est aussi fortement

dépendant du système dans lequel il évolue. En effet, généralement, les données exploitées par le système déterminent le contenu du modèle.

II.2.1 Modèles de représentation de profils utilisateurs

Différentes approches sont utilisées pour représenter les utilisateurs et structurer la connaissance à leur propos. La représentation de l'utilisateur à travers la notion de profil permet de mieux comprendre certains mécanismes cognitifs, notamment ceux permettant de percevoir le concept subjectif de la pertinence et au-delà, cibler ses besoins spécifiques dans le but d'améliorer les performances de recherche. Une approche de représentation permet d'organiser les caractéristiques de l'utilisateur afin de faciliter leur exploitation dans le processus d'accès à l'information. On distingue quatre principales approches de représentation : vectorielle, connexionniste, conceptuelle et multidimensionnelle.

II.2.1.1 Modèle vectorielle

Dans le modèle vectoriel, dite aussi ensembliste, chaque profil utilisateur se compose d'un ou plusieurs vecteurs de termes. Vecteurs de termes pondérés. Souvent représentés selon le modèle vectoriel de Salton [Salton et al., 1973]. Ces termes traduisent les centres d'intérêt de l'utilisateur. Le poids d'un terme est souvent calculé selon le schéma TF*IDF. Ce modèle est le modèle le plus utilisé. En utilisant ce modèle, nous pouvons calculer facilement la similarité cosinus d'un document quelconque avec le profil d'un utilisateur si le document est aussi représenté par un vecteur de termes. Parmi les systèmes utilisant ce modèle, nous pouvons citer Letizia [Lieberman, 1995] un système d'aide à la navigation.

Dans un cas particulier de ce modèle, les poids des termes dans le vecteur ne sont plus des valeurs réelles mais des valeurs booléennes. Chacune de ces valeurs représente la présence ou non d'un terme dans le profil.

Chapitre 2 : accès personnalisé à l'information et modélisation du profil utilisateur

Ceci est assez simple à mettre en œuvre mais présente l'inconvénient d'un manque de structuration et de l'absence de gestion des différents niveaux de généralités pouvant caractériser l'utilisateur.

La figure II.1 donne un exemple de profil utilisateur représenté par des mots clés pondérés. Ce profil contient trois centres d'intérêts : Art, Sports et Music. Chaque centre est représenté par un ensemble de termes pondérés. Music = < (Rock, 0, 15), (Symphony, 0, 87), . . . > est un extrait du l'ensemble de termes.

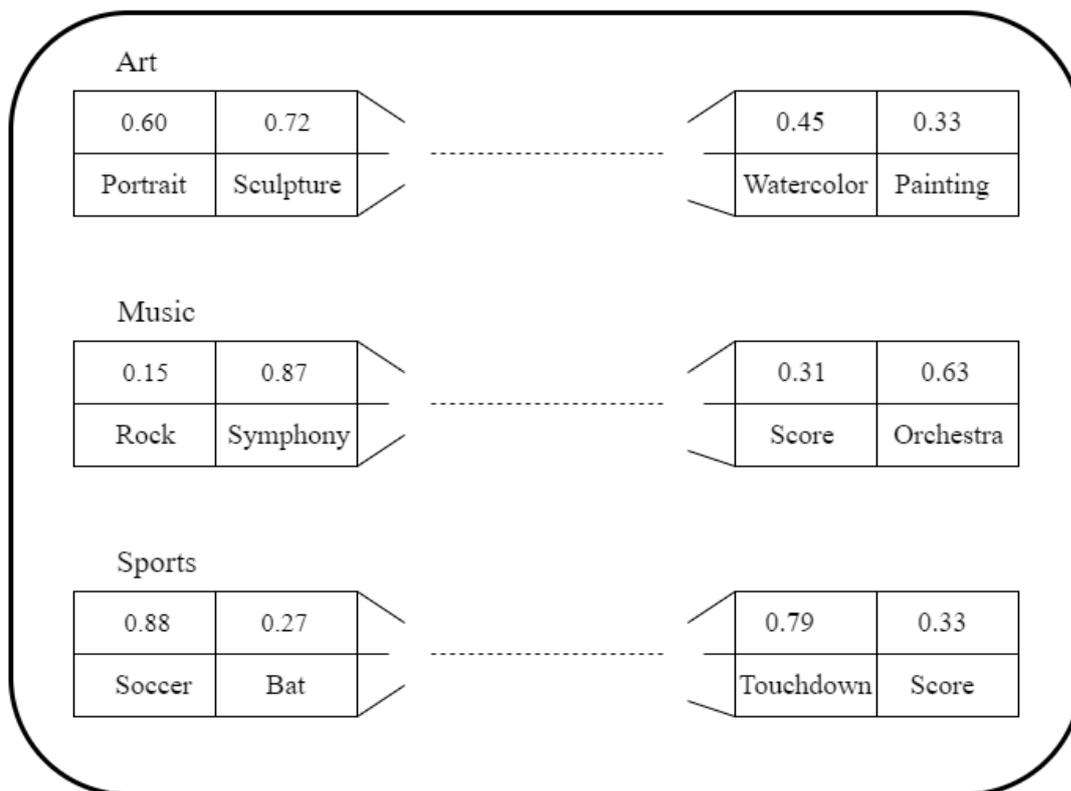


Figure II.1 Exemple de profil représenté par des mots clés

II.2.1.2 Modèle connexionniste

C'est un type de représentation basé sur l'interconnexion de nœuds représentant les termes, préférences ou documents, il offre le double avantage de la structuration et de la représentation associative permettant de considérer l'ensemble des aspects représentatifs du profil.

Cette représentation permet de résoudre les failles de la représentation ensembliste. En effet, les problèmes tels que la polysémie des termes et l'incohérence éventuelle entre les centres d'intérêts peuvent être résolus par cette représentation qui apporte de la sémantique au modèle de l'utilisateur. Plusieurs SRI personnalisés adoptent ce type de représentation ([Gentili et al., 2003], [Koutrika and Ioannidis, 2005]).

Cette représentation présente néanmoins certaines limitations. En effet, la source de données du réseau sémantique représentant le modèle de l'utilisateur n'est autre que l'historique de recherche de l'utilisateur qui est souvent assez limité.

II.2.1.3 Modèle conceptuelle

La représentation conceptuelle d'un profil utilisateur, consiste à représenter les intérêts de l'utilisateur, par un réseau de nœuds conceptuels décrivant un domaine d'intérêts de l'utilisateur et de les relier entre eux en respectant la topologie des liens, définie dans des hiérarchies (essentiellement basée sur l'utilisation d'une l'ontologie). On obtient, un profil représenté sous forme d'une hiérarchie de concepts, grâce à l'association des intérêts de l'utilisateur aux concepts des domaines de l'ontologie (Gauch) [Gauch et al., 2007].

La représentation conceptuelle est semblable à la représentation sémantique, dans le sens, où elle représente les centres d'intérêts de l'utilisateur par un réseau de nœuds conceptuels. Cependant, dans l'approche conceptuelle, les nœuds correspondent à des domaines abstraits représentant les centres d'intérêts de l'utilisateur, contrairement à l'approche sémantique où les centres d'intérêts sont représentés par un mot spécifique ou ensemble de mots relatifs.

La représentation conceptuelle peut également être assimilée à une approche ensembliste (vectorielle) du fait que les domaines sont souvent représentés comme des vecteurs de termes pondérés. Néanmoins, les termes de ces vecteurs sont regroupés pour former un domaine spécifique et non de simples mots-clés.

II.2.1.4 Modèle multidimensionnel

Cette représentation donne une description globale des utilisateurs en prenant en compte plusieurs dimensions différentes. Cette représentation a pour objectif de capturer toutes les caractéristiques informationnelles de l'utilisateur. Les propositions de standards P3P [(W3C. 2005.) pour la sécurisation des profils ont défini des classes distinguant les attributs démographiques des utilisateurs (identité, données personnelles), les attributs professionnels (employeur, adresse, type) et les attributs de comportement (trace de navigation).

Amato [**Amato and Straccia, 1996**], dans le cadre du développement d'un service de bibliothèque numérique - système EUROgather service-, représente le modèle de l'utilisateur par des dimensions (ou catégories) prédéfinies : catégorie de données personnelles, catégorie de données de la source, catégorie de données de livraison, catégorie de données de comportement et catégorie de données de sécurité.

Dans ce même cadre, Kostadinov [**Kostadinov, 2003**] a poursuivi cette classification en proposant un ensemble de dimensions ouvertes, pouvant contenir la plupart des informations susceptibles de caractériser l'utilisateur. Dans sa représentation il distingue principalement huit dimensions décrites brièvement dans ce qui suit :

* Les données personnelles

Les données personnelles sont la partie statique du profil. Elles comprennent l'identité civile de l'utilisateur (nom, prénom, numéro de sécurité sociale, etc.) ainsi que des données démographiques (âge, genre, adresse, situation familiale, nombre d'enfants, etc.)

* Le centre d'intérêt

Le centre d'intérêt exprime le domaine d'expertise de l'utilisateur. Il peut être défini par un ensemble de mots clés ou un ensemble d'expressions logiques (requêtes).

* L'ontologie du domaine

L'ontologie du domaine complète la définition du centre d'intérêt en explicitant la sémantique de certains termes ou de certains opérateurs employés par l'utilisateur dans son profil ou dans ses requêtes.

Chapitre 2 : accès personnalisé à l'information et modélisation du profil utilisateur

* La qualité attendue

La qualité est un des facteurs clés de la personnalisation, elle permet d'exprimer des préférences extrinsèques comme l'origine de l'information, sa précision, sa fraîcheur, sa durée de validité, le temps nécessaire pour la produire ou la crédibilité de sa source. Les attributs de cette dimension expriment la qualité attendue ou espérée par l'utilisateur.

* La customisation

La customisation concerne d'abord tout ce qui est lié aux modalités de présentation des résultats en fonction de la plateforme, de la nature et du volume des informations délivrées, des préférences esthétiques ou visuelles de l'utilisateur.

* La sécurité

La sécurité est une dimension fondamentale du profil. Elle peut concerner les données que l'on interroge ou modifie, les informations que l'on calcule, les requêtes utilisateurs elles-mêmes ou les autres dimensions du profil. La sécurité du processus exprime la volonté de l'utilisateur à cacher un traitement qu'il effectue.

* Le retour de préférences

On désigne par ces termes ce qu'on appelle communément le « feedback » de l'utilisateur. Cette dimension regroupe l'ensemble des informations collectées sur l'utilisateur.

* Les informations diverses

Certaines applications demandent des informations spécifiques ne pouvant être incluses dans aucune des dimensions précédentes comme par exemple la bande passante attribuée au gestionnaire du profil. Pour cette raison l'utilisateur a la possibilité de rajouter ce type de préférences dans la partie diverse du profil et de décrire leurs utilisations.

Afin de mener une réflexion globale sur la personnalisation de l'information dans un environnement à grande échelle aussi bien dans un environnement de RI que de BD. Les travaux

Chapitre 2 : accès personnalisé à l'information et modélisation du profil utilisateur

menés dans ce cadre par Kostadinov [Kostadinov, 2007] ont abouti à la proposition d'un modèle utilisateur générique englobant les notions de profil, de préférence et de contexte.

Un profil utilisateur représente l'ensemble des informations décrivant l'utilisateur. Un contexte représente les données décrivant l'environnement d'interaction entre un utilisateur et un système. Une préférence est une expression permettant de hiérarchiser l'importance des informations dans un profil ou un contexte.

II.2.2 Construction du modèle de l'utilisateur

L'approche de construction dépend fortement de la représentation choisie pour le profil de l'utilisateur. La démarche de construction est commune à tous les systèmes et s'effectue en deux étapes : l'acquisition et la collecte des données utilisateur, puis la construction proprement dite du profil comme le montre la figure II.2.

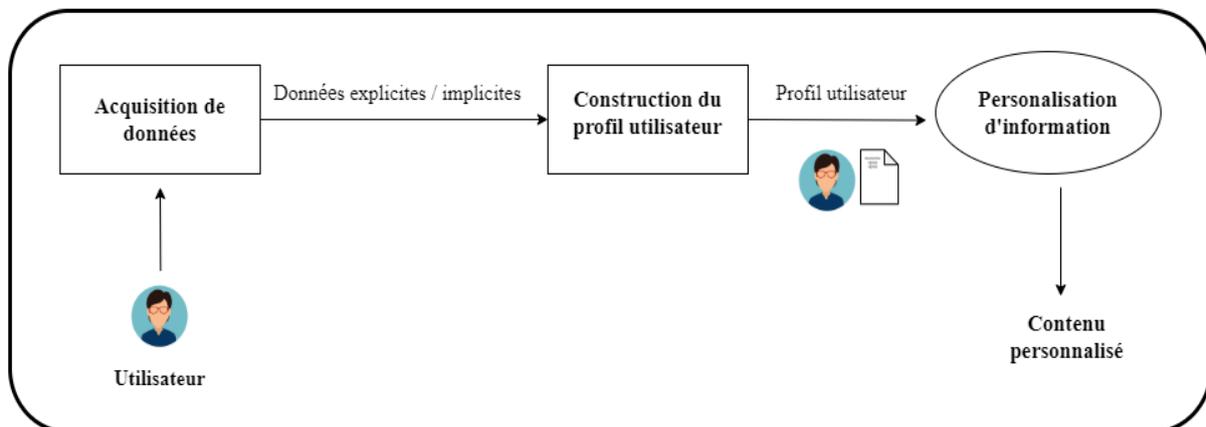


Figure II.2 les phases de construction du profil utilisateur

II.2.2.1 Acquisition de données utilisateur

La phase d'acquisition des données utilisateur consiste à collecter les informations qui serviront à l'instanciation du modèle de l'utilisateur. Nous distinguons deux modes d'acquisition : acquisition explicite et acquisition implicite. Dans l'approche explicite, les informations sont

directement obtenues de l'utilisateur. Dans l'approche implicite, ce sont les données de comportement de l'utilisateur qui sont exploitées. Nous détaillons ces approches dans ce qui suit.

- **Acquisition explicite des données**

La technique d'acquisition explicite des données est une technique simple, qui consiste à interroger l'utilisateur, pour lui demander des informations personnelles, démographiques et/ou ses intérêts (Gauch) [Gauch et al., 2007] via des formulaires par exemple My Yahoo. Ou bien en considérant le feedback explicite de l'utilisateur exprimé lors de son interaction avec le système, par exemple, fournir une note sur une échelle de valeurs prédéfinies (notes que les internautes indiquent sur des produits qu'ils achètent sur Internet par exemple), faire une action de recommandation (article qu'un utilisateur recommande à un autre utilisateur par exemple), exprimer une opinion polarisée sur un objet (exemple du bouton « j'aime » sur Facebook), etc.

L'avantage des approches explicites est que les profils ainsi construits sont plus précis que ceux obtenus par acquisition implicite. Cependant, ce procédé d'acquisition présente le principal inconvénient de pouvoir induire au désintéressement de l'utilisateur lors de la saisie de données (Wu) [Wu et al., 2008]. De plus, la construction du modèle de l'utilisateur étant fortement liée au mode d'acquisition de données, elle se retrouve fortement liée au degré d'implication de l'utilisateur. En Effet, si l'utilisateur ne fournit pas les informations, aucun modèle ne sera construit.

- **Acquisition implicite des données**

Dans cette approche, le système acquiert les informations nécessaires pour construire les profils utilisateurs implicitement en surveillant ou traçant leurs activités. Les activités que peut avoir un utilisateur peuvent correspondre à l'utilisation de moteurs de recherche à travers la soumission de requêtes et la sélection de documents, la navigation sur le web, les contenus publiés sur le web ou sur les réseaux sociaux (annotations, commentaires) ou encore l'utilisation d'applications dans le contexte de sa recherche (applications de bureautique, éditeurs de texte, ...). Une fois la collecte de données effectuée, le profil de l'utilisateur est dérivé.

Cette approche présente le principal avantage de ne pas impliquer l'utilisateur et de ne pas lui imposer l'émission de jugements, ni un effort d'attention particulier lors de sa recherche. Néanmoins, la difficulté dans cette technique est de définir un processus d'interprétation du comportement observé dans un contexte d'application spécifique.

II.2.2.2 Techniques de construction du profil de l'utilisateur

La construction du profil utilisateur consiste à organiser et extraire les éléments qui constituent le profil à partir des données de l'utilisateur collectées lors de l'étape précédente, de l'étape précédente. On distingue différentes techniques de construction du profil utilisateur, basées sur l'analyse statistique de termes, le filtrage collaboratif et les techniques d'apprentissage [Gauch et al., 2007]

- **Analyse statistique de termes**

Cette approche permet d'analyser le contenu d'un document et d'extraire les mots clés significatifs qui décrivent son contenu. Ces mots clés sont stockés pour être utilisés ensuite afin de comparer des éléments entre eux ou avec les préférences de l'utilisateur lorsqu'elles sont exprimées sous la forme de mots clés.

Il existe différentes structures de stockage et de représentation des mots clés, on distingue trois principales techniques : l'extraction des termes, l'extraction de réseaux de termes et l'extraction de concepts.

Extraction d'ensemble de termes : se base sur des techniques d'extraction des termes à partir des données visitées par l'utilisateur. Ces termes vont servir dans l'algorithme d'apprentissage du modèle de l'utilisateur.

Parmi les systèmes appliquant cette approche de construction, on peut citer les systèmes WebMate [Chen and Sycara, 1998] et Alipes [Widyantoro et al., 1999].

Chapitre 2 : accès personnalisé à l'information et modélisation du profil utilisateur

Extraction de réseaux de termes : les termes sont extraits des documents jugés par l'utilisateur et représentés sous forme de réseau de nœuds. Afin de construire le profil utilisateur, il est nécessaire d'exploiter des relations préexistantes entre les termes et les concepts. Le système SiteIF [Stefani and Strappavara, 1998] utilise cette technique en se basant sur le dictionnaire WordNet.

Extraction de concepts : concerne principalement les profils utilisateurs représentés par une hiérarchie de concepts pondérés, elle utilise une taxonomie de concepts de référence comme profil de base. Les principales ressources sémantiques utilisées sont ODP 3 un annuaire de concepts hiérarchique open source et la hiérarchie de concepts Yahoo, L'approche de construction présente de manière générale les deux étapes suivantes :

- (1) identification des concepts et niveaux de l'ontologie à exploiter
- (2) extraction des centres d'intérêts de l'utilisateur par analogie aux concepts de l'ontologie.

● Filtrage collaboratif

Cette approche consiste à comparer les profils des utilisateurs entre eux afin de trouver des similarités dans leur comportement et dans leurs préférences et de faire des recommandations sur la base de ces similarités. Le filtrage collaboratif est utilisé le plus souvent dans les systèmes de recommandation d'éléments lorsque les profils des utilisateurs sont composés par un ensemble de couples (objet, pertinence de l'objet).

Ce type de filtrage est caractérisé par sa flexibilité lors du passage à l'échelle (...) et l'indépendance du format des objets manipulés

● Techniques d'apprentissage

Le principe de base de ces techniques est l'étude du comportement de l'utilisateur et la classification de ses caractéristiques ou celles des objets recherchés. Ces techniques sont

caractérisées par leur fraîcheur et l'exactitude des données dérivées. Leur inconvénient est lié à la complexité temporelle des algorithmes utilisés. Des exemples de telles techniques sont les réseaux de neurones, les méthodes de classification (raisonnement par cas, classificateurs bayésiens...), les règles d'association.

II.2.3 Évolution du profil utilisateur

La gestion de l'évolution du profil utilisateur est un processus complémentaire à sa construction. Elle désigne l'adaptation du profil à la variation des intérêts et aux variations des besoins en information de l'utilisateur au cours du temps [Zemirli, 2008]. Les intérêts de l'utilisateur peuvent changer et devenir non pertinents dans le temps, un intérêt jugé pertinent dans une période de temps peut devenir obsolète dans les périodes suivantes.

L'adaptation du profil utilisateur à cette évolution implique des changements au niveau des intérêts décrits dans le profil qui conduisent éventuellement à la suppression d'intérêts existants, à l'émergence de nouveaux intérêts et dans tous les cas au (re)calcul des poids des intérêts. L'évolution du profil utilisateur peut s'effectuer lors de sa modélisation par deux modèles :

Le profil à court terme qui contient les informations récentes liées à l'activité courante de l'utilisateur (dernière fenêtre temporelle). Ses informations sont partiellement représentées par le sujet de la requête, Le but fondamental de l'évolution du profil à court terme est d'améliorer la précision de recherche en utilisant le profil le plus utile et approprié à la requête et non bruité par des centres d'intérêts hors contexte de recherche.

Le profil à long terme qui contient des informations qui reflètent les besoins à long terme de l'utilisateur, généralement cumulées dans le temps, Ce profil peut être exploitable dans le but d'améliorer la recherche pour toute requête soumise par l'utilisateur.

II.3 Les approches de personnalisation

La personnalisation du processus d'accès à l'information consiste à intégrer ou exploiter le profil utilisateur dans la chaîne d'accès à l'information. Le but fondamental des modèles d'accès personnalisé à l'information est de restituer, en haut de la liste des résultats, des documents qui intéressent l'utilisateur dans sa recherche, en d'autres termes qui semblent les plus similaires à son profil. Nous présentons dans cette section les différentes approches qui consistent à intégrer le profil dans au moins l'une des phases du processus d'accès à l'information telle que l'enrichissement des requêtes, le filtrage des résultats et le reclassement de résultats.

II.3.1 L'enrichissement des requêtes

Ce service consiste à enrichir la requête de l'utilisateur par un ensemble de prédicats contenus dans son profil. L'enrichissement des requêtes exploite donc le profil de l'utilisateur pour reformuler sa requête en y intégrant des éléments définis dans son profil (préférences, centre d'intérêt, etc.). La méthode la plus aboutie est celle de Koutrika [Koutrika and Ioannidis, 2004]. Dans cette méthode, le profil de l'utilisateur est composé d'un ensemble des prédicats pondérés. Le poids du prédicat exprime son intérêt relatif pour l'utilisateur. Il est spécifié par un réel compris entre 0 et 1.

II.3.2 Le filtrage des résultats

Dans la tâche de filtrage [Belkin and Croft, 1992], un profil utilisateur doit décrire les préférences de l'utilisateur. Un tel profil est alors comparé aux documents entrants afin d'essayer de déterminer ceux qui pourraient être intéressants pour cet utilisateur particulier.

A l'arrivée de nouveaux documents, le système utilise le profil utilisateur pour choisir les documents pouvant potentiellement l'intéresser. Par un processus de retour de pertinence « Relevance feedback », l'utilisateur n'indique pas seulement les documents pertinents mais aussi

les documents non-pertinents (figure II.3). Le système utilise ces informations pour ajuster la description du profil utilisateur qui reflètera les nouvelles préférences.

L'avantage du filtrage des résultats est sa simplicité parce qu'il ne nécessite aucune modification du fonctionnement des fournisseurs d'information. Tout le traitement est fait après l'exécution de la requête. Les inconvénients sont le volume de données échangées entre le serveur et le client et le risque d'élimination d'éléments pertinents.

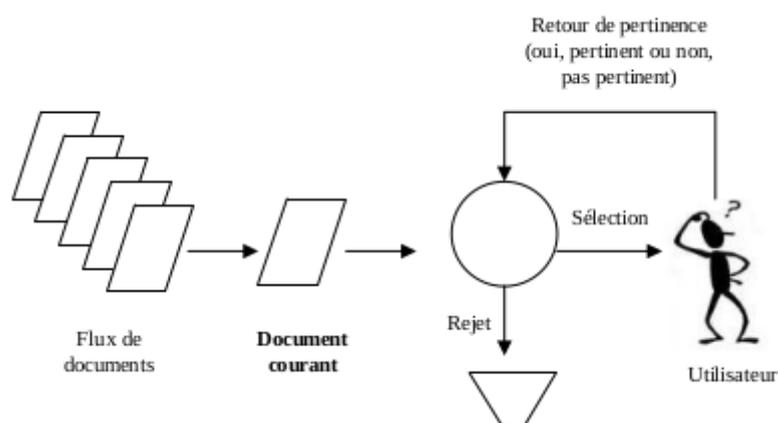


Figure II.3 Représentation d'un système de filtrage

II.3.3 Reclassement de résultats

Dans un système de RI personnalisé qui utilise cette approche, le système envoie une requête à un moteur de recherche, reçoit des résultats et puis trie les résultats selon leurs similarités avec le profil d'utilisateur.

Dans les travaux de Gauch [Gauch et al., 2003] qui utilise le modèle ontologie pour représenter des profils utilisateurs. Ils utilisent le profil utilisateur pour trier à nouveau les résultats obtenus par le méta-moteur de recherche ProFusion. Pour chaque résultat r (document retourné) de ce méta-moteur, ils calculent un nouveau score new_wt_r en utilisant le résultat original du moteur, la similarité entre le résultat et les concepts correspondant, et l'intérêt de l'utilisateur avec ces concepts (les poids de ces concepts) :

$$new - wt_r = wt_r \left(0,5 + \frac{1}{4} \sum_{l=1}^4 u_{crl} \right) \quad (2.1)$$

Dans cette formule, wt_r est le score original calculé par le moteur de recherche pour le résultat r , u_{crl} est l'intérêt de l'utilisateur avec le concept crl dans son profil, et crl est le l eme concepts parmi les concepts les plus similaires avec le résultat r . Le nouveau score new_wt_r sera utilisé pour trier à nouveau les documents. Parmi les autres travaux qui utilisent les profils utilisateurs pour le reclassement des résultats de recherche, nous pouvons citer ([Rohini and Ambati, 2005], [Shen et al., 2005]).

L'avantage de cette approche est qu'elle n'exclue pas les éléments du résultat, l'utilisateur vas surement trouver ce qu'il l'intéresse, mais elle est peu utilisée dans les systèmes personnalisés car elle nécessite le calcul du nouveau rang de chaque élément du résultat ce qui demande beaucoup de temps.

II.4 CONCLUSION

Dans ce chapitre, nous avons présenté les principaux systèmes de personnalisation de la recherche d'information dont le point commun est la prise en compte du composant utilisateur dans le processus de recherche. Nous avons passé en revue les différentes approches et techniques de modélisation du profil de l'utilisateur, à savoir sa représentation, sa construction et son évolution au cours du temps ainsi que son intégration dans les systèmes de recherche d'information.

Chapitre 2 : accès personnalisé à l'information et modélisation du profil utilisateur

Ce chapitre permet de conclure que la modélisation du profil utilisateur constitue un élément essentiel dans le développement d'un système d'accès personnalisé performant dont l'efficacité dépend fortement de la précision des profils. Parmi les techniques qui nous permettent de définir au mieux ce profil on distingue les réseaux de neurones.

Nous présentons dans le prochain chapitre les réseaux de neurones.

Chapitre 03 : Apprentissage profond “ Deep Learning”.

III.1 Introduction

Des améliorations considérables ont été observées ces dernières années dans des domaines d'applications tels que la vision par ordinateur, la reconnaissance de la parole, d'image et la traduction automatique. Ces progrès ont été essentiellement stimulés par les récentes avancées en apprentissage automatique, et plus particulièrement en apprentissage profond, basé sur les réseaux de neurones.

Dans ce chapitre, nous présentons un ensemble de généralités et de concepts relatifs à l'apprentissage automatique et les réseaux de neurones. En premier lieu nous donnons quelques définitions de l'apprentissage automatique et nous présentons les différents types et modèles. Après cela, nous présentons un ensemble de généralités et de concepts relatifs aux réseaux de neurones, leur fonctionnement. Puis, et avant de conclure, nous donnons l'architecture des réseaux de neurones, quelques algorithmes d'apprentissage et la classification automatique de textes.

III.2 Définitions de l'apprentissage automatique

L'apprentissage automatique regroupe tellement de concepts différents et variés qu'il est difficile d'en donner une définition unique, Pour cela nous exposant trois définitions différentes.

Définition 1 : L'apprentissage automatique est un domaine de l'intelligence artificielle qui fait référence au développement à l'analyse et à l'implémentation de méthodes automatisables permettant à une machine d'évoluer grâce à un processus d'apprentissage, et ainsi de remplir des tâches qu'il est difficile ou impossible de satisfaire par des moyens algorithmiques plus classiques [Hammoud, 2014].

Définition 2 : L'apprentissage automatique est une discipline scientifique utilisée en intelligence artificielle. Il s'agit d'algorithmes (procédures traduites en langages informatiques)

qui analysent un ensemble de données afin de déduire des règles constituant de nouvelles connaissances et permettant d'analyser de nouvelles situations [Ammar, 2007].

Définition 3 : L'apprentissage automatique (ou apprentissage artificiel) est l'étude des algorithmes qui permettent aux programmes de s'améliorer automatiquement par expérience [Tellie, 2010].

III.2.1 Types d'apprentissage automatique

Il existe plusieurs types d'apprentissage automatique, qui se distinguent essentiellement par leur objectif. Parmi les types d'apprentissage automatique les plus largement adoptés sont :

III.2.1.1 L'apprentissage supervisé

Cette approche a pour objectif de concevoir un modèle reliant des données d'apprentissage à un ensemble de valeurs de sortie (un comportement) [Amraoui, 2011]. Un superviseur (expert) est employé pour étiqueter correctement des exemples (les données d'apprentissage ou d'entraînement). L'apprenant (ordinateur ou machine) doit alors trouver ou approximer la fonction qui permet d'affecter la bonne étiquette à ces exemples [Krief and Benmammar, 2006].

La construction et la génération d'un modèle en apprentissage supervisé passe par deux phases :

- **Phase d'apprentissage (d'entraînement)**

Dans cette phase l'algorithme d'apprentissage reçoit en entrée des exemples d'apprentissage (les documents d'entraînement) étiquetés et produit un modèle de prédiction le plus performant possible, c'est-à-dire le modèle qui produit le moins d'erreurs.

- **Phase de test**

Dans cette phase le modèle obtenu lors de la phase d'apprentissage doit être capable de prédire l'étiquette d'un nouvel exemple en fonction des valeurs d'entrées.

On distingue deux types de problèmes de l'apprentissage supervisé :

- **La classification :**

Consiste à identifier les classes d'appartenance de nouveaux objets à partir d'exemples antérieurs connus, la variable à prédire peut donc prendre des valeurs discrètes appelées classes [Laur, 2017] (exemple : bon, moyen, mauvais).

- **La régression**

Utilisée lorsqu'il s'agit de prédire une variable continue, qui peut donc prendre un nombre infini de valeurs (exemple : température, gains monétaires, etc). Par exemple, pour une expérience de biochimie, on pourrait vouloir prédire le taux de résistance d'un organisme en fonction des taux de différentes substances qui lui sont administrées [Laur, 2017].

III.2.1.2 L'apprentissage non supervisé

Aucun expert n'est disponible. Il vise à concevoir un modèle structurant l'information. La différence ici est que les comportements (ou catégories ou encore les classes) des données d'apprentissage ne sont pas connus, c'est ce que l'on cherche à trouver [Larochelle, 2008].

L'apprentissage non supervisé ne nécessite pas l'étiquetage des données de la base d'apprentissage. Son objectif est donc de trouver tout seul des points communs parmi ses données d'entrée.

III.2.1.3 Apprentissage par Renforcement

Les données en entrée sont les mêmes que pour l'apprentissage supervisé, cependant l'apprentissage est guidé par l'environnement sous la forme de récompenses ou de pénalités données en fonction de l'erreur commise lors de l'apprentissage. [Laroche, 2008].

III.2.1.4 Apprentissage semi-supervisé

Ce type d'apprentissage utilise un ensemble de données étiquetées et non-étiquetées. Il se situe ainsi entre l'apprentissage supervisé qui n'utilise que des données étiquetées et l'apprentissage non-supervisé qui n'utilise que des données non-étiquetées. Ainsi, Il a été démontré que l'utilisation de données non-étiquetées, en combinaison avec des données étiquetées, permet d'améliorer significativement la qualité de l'apprentissage. Un autre intérêt provient du fait que l'étiquetage de données nécessite l'intervention d'un expert [Laroche, 2008].

III.3 L'apprentissage profond

III.3.1 Définition

L'apprentissage profond (ou Deep Learning en anglais) est une branche de l'apprentissage automatique (ou Machine Learning en anglais). [Patterson and Gibson, 2017].

La figure III.1 illustre l'imbrication entre ces deux branches avec la branche de l'intelligence artificielle (IA).

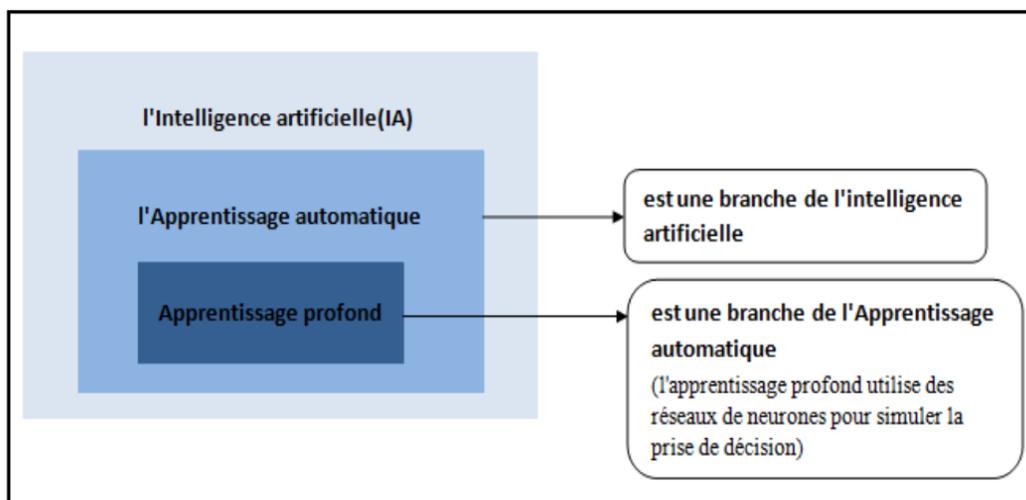


Figure III.1 La relation entre les trois branches l'intelligence artificielle(IA), (ML) et (DL).

L'apprentissage profond a été introduit dans l'étude des réseaux de neurones ; inspirée à l'origine par une étude du fonctionnement physiologique du cerveau. Il permet d'apprendre et de traiter des données complexes et tente également de résoudre des tâches complexes [Nene, 2017].

L'apprentissage profond permet à des modèles composés de plusieurs couches de traitement d'apprendre des représentations des données avec de multiples niveaux d'abstraction.

Il utilise des algorithmes de réseaux de neurones avec un nombre important de couches, les premières couches traversées par les données faisaient ce travail de préparation des données pour les couches plus profondes. [Patterson and Gibson, 2017] .

Dans ce qui suit nous présentons le principe du fonctionnement des réseaux de neurones et leurs architectures.

III.4 Les réseaux de neurones

Les réseaux de neurones sont un modèle d'apprentissage automatique suivant un apprentissage supervisé et non supervisé. Ce dernier est conçu pour reconnaître des modèles de comportement dans les données. L'objectif est d'apprendre le modèle qui permet d'encoder toutes les données du monde réel (image, son, texte) en un vecteur numérique. Ainsi, ils permettent de solutionner des problèmes jadis qualifiés de complexes.

La Figure III.2 illustre les parties équivalentes entre un neurone biologique et un neurone formel.

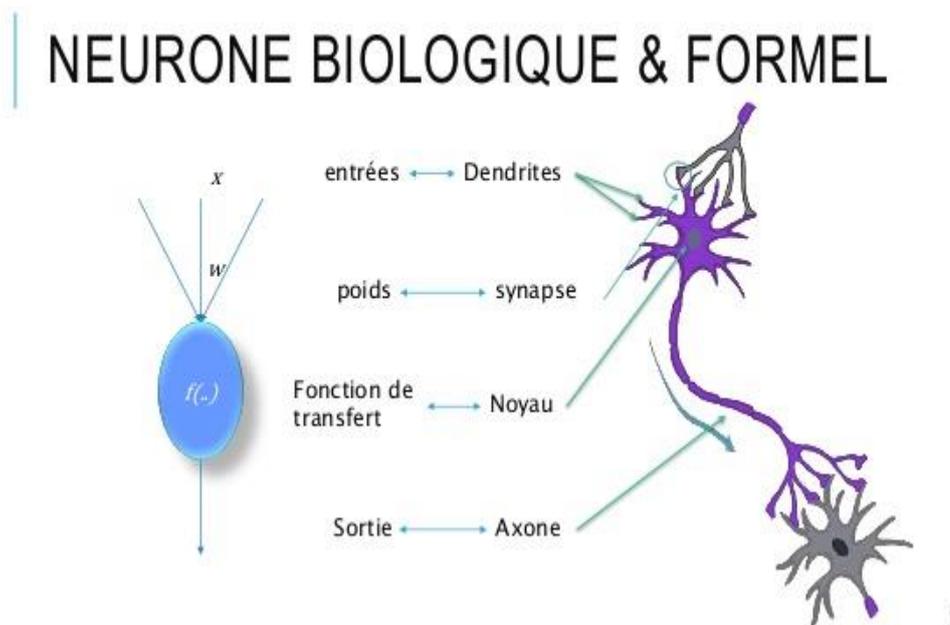


Figure III.2 Equivalence entre un neurone biologique et un neurone formel

III.4.1 Neurone formel

Un neurone formel dite aussi artificiel est une représentation mathématique et informatique d'un neurone biologique, un neurone artificiel est une fonction algébrique non linéaire, paramétrée

Chapitre 3 : Apprentissage profond « deep learning »

et à valeurs bornées. La figure III.3 représente la structure générale d'un neurone artificiel, qui se caractérise comme suit :

- Ses entrées ($X_1, X_2, \dots, X_i, \dots, X_n$) peuvent être collectées depuis un corpus qu'on souhaite traiter ou les sorties d'autres neurones.
- La somme pondérée de ses entrées définissant le prétraitement (combinaison linéaire) effectué sur les entrées comme le montre l'équation :

$$\sum W_i \cdot X_i + b \quad (3.1)$$

Où w_i est le poids synaptique attaché à l'entrée i et le b désigne le seuil d'activation.

- Fonction d'activation f , définissant l'état interne du neurone en fonction de son entrée totale.

Y_1 est la sortie calculant la sortie du neurone en fonction de son état d'activation [Bendaoud, 2014].

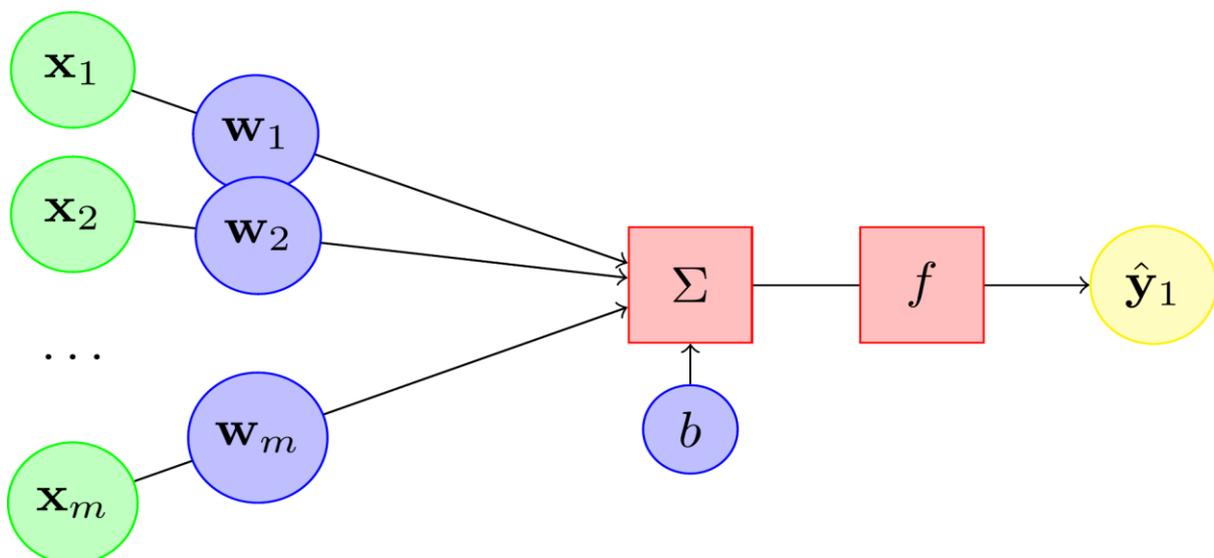


Figure III.3 Représente la structure générale d'un neurone formel.

III.4.1.1 Définition d'un réseau de neurones artificiels

- **Définition 1**

Les réseaux de neurones artificiels sont des réseaux fortement connectés de processeurs élémentaires fonctionnant en parallèle. Chaque processeur élémentaire calcule une sortie unique sur la base des informations qu'il reçoit [Rosenfeld et al., 1988] .

- **Définition 2**

Un réseau de neurones peut être considéré comme un modèle mathématique de traitement réparti, composé de plusieurs éléments de calcul non-linéaires (les neurones), opérant en parallèle et connectés entre eux par des poids [Ammar, 2007] .

Dans un modèle de neurones chaque couche a son rôle bien déterminé, elle fait un traitement indépendant en transmettant le résultat à la couche suivant du modèle. L'information donnée au réseau va donc se propager couche par couche ; de la couche d'entrée à la couche de sortie ; en passant soit par aucune, soit par plusieurs couches intermédiaires ; dites couches cachées (hidden layout).

Généralement chaque neurone dans une couche est connecté à tous les neurones de la couche suivante ainsi la couche précédente, excepté pour les couches d'entrées et de sorties. Exemple : nous pourrions avoir un neurone de la couche d'entrée qui connecte juste vers un seul neurone de la couche suivante (couche cachée). Par conséquent ce neurone de la couche d'entrée s'intéresse uniquement à une tâche précise. De cela nous distinguons trois type de couches qui seront représentés dans ce qui suit :

- **Couche d'entrée** : les neurones de cette couche reçoivent les valeurs d'entrées du réseau et les transmettent aux neurones cachés. Chaque neurone reçoit une valeur (aucun traitement se fait au niveau de la couche d'entrée).
- **Couches cachées** : chaque neurone de cette couche reçoit l'information de plusieurs couches précédentes, effectue la sommation pondérée par les poids selon la formule 3.1.

Le résultat de l'équation est transformé selon sa fonction d'activation. Par la suite, il envoie cette réponse aux neurones de la couche suivante.

- **Couche de sortie** : elle joue le même rôle que les couches cachées, la seule différence entre ces deux types de couches est que la sortie des neurones de la couche de sortie n'est connecté à aucun autre neurone [Ammar, 2007] .

III.4.2 Fonction d'activation

La fonction d'activation sert à convertir le résultat de la somme pondérée des entrées d'un neurone en une valeur de sortie. Quand le résultat de la somme pondérée dépasse le seuil, l'argument de cette fonction devient positif ou nul sinon est considéré négatif. Finalement, si le résultat de la somme pondérée est :

- En dessous du seuil, le neurone est inactif.
- Aux alentours du seuil, le neurone est considéré en phase de transition.
- Au-dessus du seuil, le neurone est actif.

Il existe plusieurs types de fonctions d'activation qui sont utilisés dans des réseaux de neurones artificiels, nous présentons ci-dessous les fonctions les plus utilisés :

III.4.2.1 La fonction Sigmoidale

Est une fonction continue. Elle est utilisée lorsque les valeurs à prédire par le RN sont comprises dans l'intervalle $[0, 1]$. Son but est d'exprimer sa valeur de sortie sous forme d'une probabilité, si la valeur en entrée est un très grand nombre positif, la fonction convertira cette valeur en une probabilité de 1. A l'inverse, si la valeur en entrée est un très petit nombre négatif, la fonction convertira cette valeur en une probabilité de 0 [Chaoui, 2002] .

La fonction Sigmoidale est définie par :

$$f(x) = \frac{1}{1+e^{-x}} \quad (3.2)$$

Où x est un nombre réel.

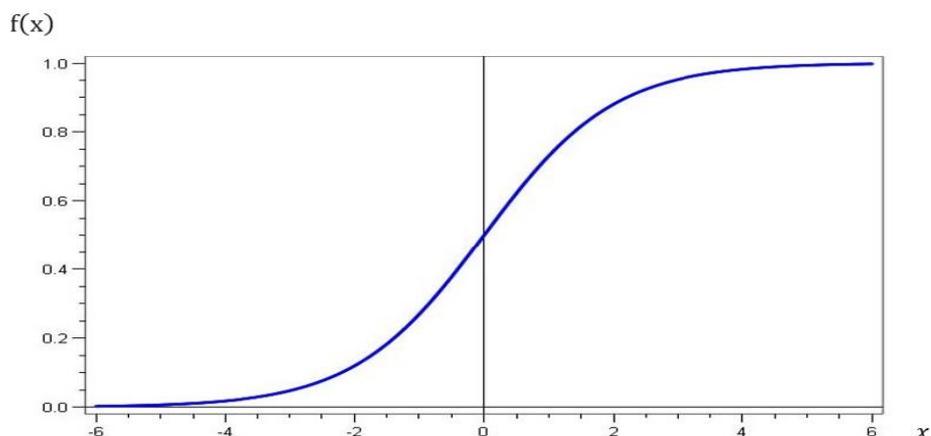


Figure III.4 La fonction sigmoïde

III.4.2.2 La fonction Tangente Hyperbolique (Tanh)

Similaire à la fonction Sigmoïde. La différence avec la fonction Sigmoïde est que la fonction Tanh produit un résultat compris entre -1 et 1. Elle produit généralement de meilleurs résultats que la fonction Sigmoïde en raison de sa symétrie, où ses petites entrées négatives tendent vers -1 et ses grandes entrées positives tendent vers 1. Idéale pour les perceptrons multicouches, en particulier, pour les couches cachées.

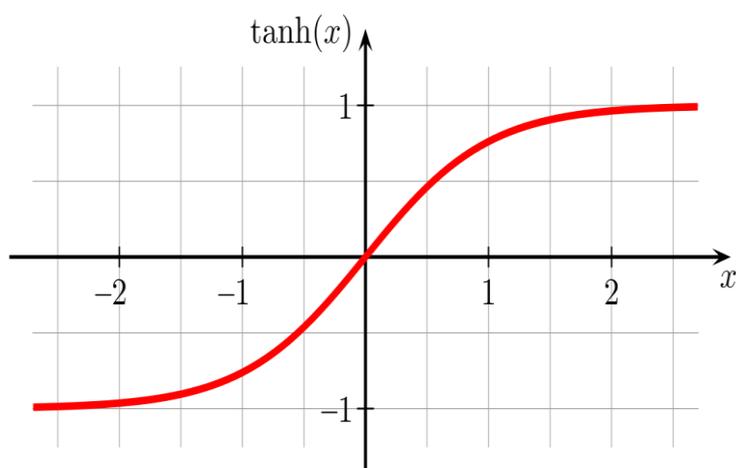


Figure III.5 La fonction Tangente Hyperbolique.

III.4.2.3 Fonction Relu

La fonction Relu (Unité de Rectification Linéaire) est créé pour palier au problème de saturation (est la situation où le gradient reste presque nul après chaque période de temps durant le processus d'apprentissage) des deux fonctions (Sigmoidé et Tanh). Relu désigne la fonction réelle non-linéaire interprétée par la formule :

$$f(x) = \max(0, x). \quad (3.3)$$

Si l'entrée est négative alors la sortie est 0, si l'entrée est positive alors la sortie est x. Cette fonction d'activation augmente considérablement la convergence du réseau et ne sature pas.

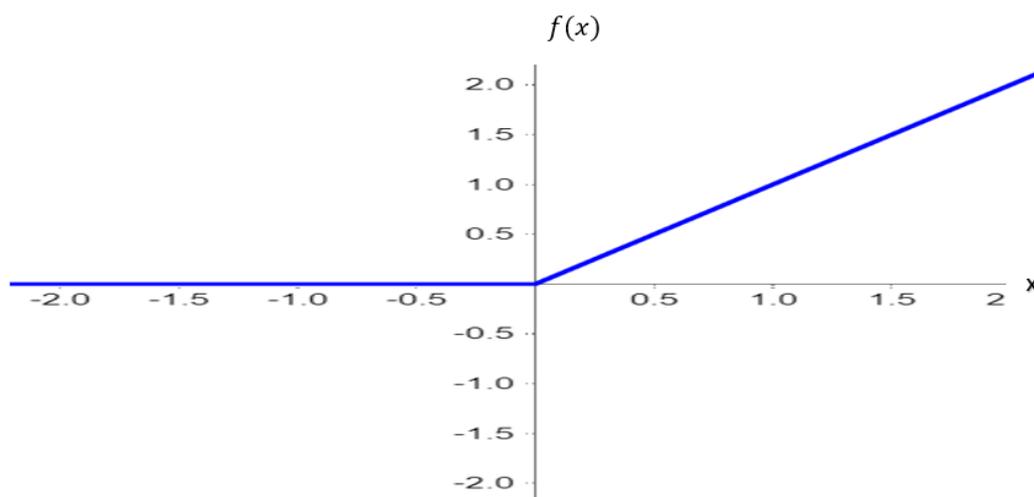


Figure III.6 La fonction relu.

III.4.2.4 Fonction SoftMax

La fonction SoftMax peut gérer plusieurs entrées. Elle attribue des probabilités à chaque entrée. Elle écrase les sorties de chaque unité entre 0 et 1 comme la fonction Sigmoidé. Mais elle divise également chaque sortie de manière à ce que la somme totale des sorties soit égale à 1. Cette contrainte supplémentaire permet de faire converger l'apprentissage plus rapidement qu'il ne le ferait autrement.

Chapitre 3 : Apprentissage profond « deep learning »

Essentiellement utilisée (mais pas uniquement) pour des tâches de classification. Permet de construire des réseaux de neurones avec plusieurs sorties normalisées ce qui la rend particulièrement adapté à la création de classifications par les réseaux de neurones avec des sorties probabilistes. Autrement dit, elle est souvent utilisée dans la dernière couche d'un réseau de neurones utilisé comme classifieur multi-classes. Elle permet d'attribuer des probabilités à plusieurs classes qui s'additionnent à 1.

La fonction softmax est interprétée par la formule :

$$\sigma(Z) = \frac{e^{z_j}}{\sum_{k=1}^k e^{z_k}} \quad (3.4)$$

Où z est un vecteur des entrées du calque de sortie (si vous avez 10 unités de sortie, il y a 10 éléments dans z). Et encore une fois, j indexe les unités de sortie, donc $j = 1, 2, \dots, k$.

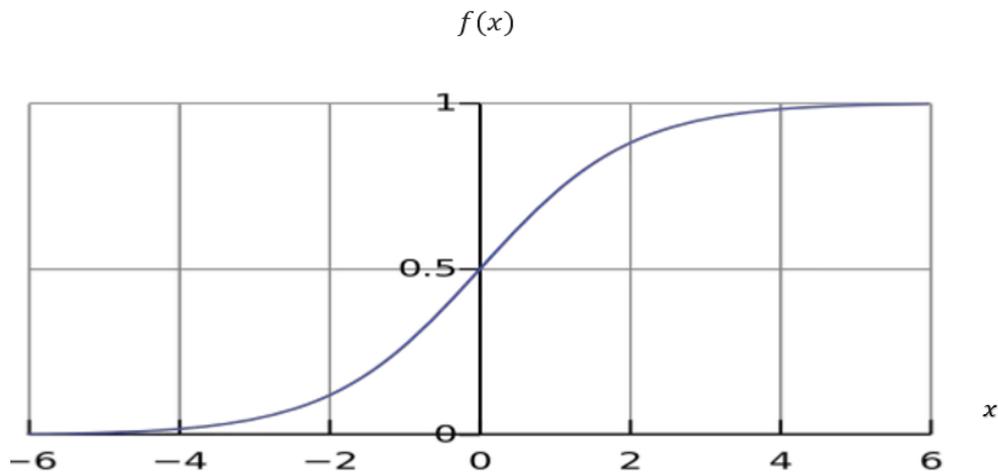


Figure III.7 : La fonction SoftMax

III.4.3 Fonctionnement des réseaux de neurones

Un réseau de neurones combine plusieurs couches de traitement en utilisant des éléments simples fonctionnant en parallèle et inspirés du système nerveux biologique. Il se compose d'une couche d'entrée, d'une ou de plusieurs couches cachées et d'une couche de sortie. Les

couches sont interconnectées par des nœuds, ou neurones, chaque couche utilisant la sortie de la couche précédente en guise d'entrée. Il peut apprendre à partir de données. Il peut ainsi être entraîné sur de nombreux exemples en vue de reconnaître des modèles au niveau de l'image ou de texte par exemple, aussi de classer des données et prédire les cours du marché, on distingue deux phases : la propagation directe qui consiste à propager l'information vers la sortie et la retro-propagation pour modifier les poids de connexion.

Dans ce qui suit nous présentons ces deux phases.

III.4.3.1 La propagation direct

Le réseau de neurones une fois entraîné, calcule sa sortie en propageant l'information et en effectuant les calculs depuis la couche d'entrée jusqu'à la couche de sortie (la propagation en anglais forward-propagation), en appliquant couche par couche les fonctions des neurones. Pour obtenir une bonne valeur de sortie, il est donc nécessaire de configurer les paramètres des neurones (les poids) pour qu'à chaque instance des données proposées, une valeur convenable lui soit associée.

III.4.3.2 La rétro-propagation

La rétro-propagation (en anglais back-propagation) consiste dans un premier temps à circuler vers l'avant les données d'entrées jusqu'à l'obtention d'une entrée calculée par le réseau. Puis dans un second temps, elle compare la sortie calculée à la sortie réelle connue via une fonction de coût. Les poids sont modifiés de telle sorte qu'à la prochaine itération, l'erreur commise entre la sortie calculée est minimisée. En prenant en considération la présence des couches cachées, l'erreur est rétro-propagée vers l'arrière jusqu'à la couche d'entrée tout en modifiant la pondération des neurones (toujours dans le but de minimiser cette erreur). Le processus est répété sur tous les exemples jusqu'à obtenir une erreur de sortie considérée comme négligeable.

Chapitre 3 : Apprentissage profond « deep learning »

En plus du bon choix de l'architecture d'un réseau de neurones, l'ajustement des paramètres de chacun de ses neurones est effectué, jusqu'à l'obtention des sorties attendues. Cette étape est dite « entraînement ».

Lors de l'entraînement du réseau de neurones, un ensemble de paramètres est entraîné ou ajusté (par exemple les poids des neurones) suivant un ensemble de données d'entraînement.

Un entraînement valide d'un réseau de neurones nécessite plusieurs epochs, telle que, une epoch correspond au parcours complet de l'ensemble de données. Lors de chaque epoch, un taux d'apprentissage (learning rate) est fixé et permet de définir avec quelle proportion les poids des neurones sont mis à jour. Celui-ci est choisi de telle sorte à ce qu'il soit grand lors des premières epochs (au début de l'entraînement), puis réduit progressivement lors des epochs suivantes.

Il existe plusieurs algorithmes d'entraînement qui permettent de mettre à jour les poids associés à chaque neurone au sein du réseau dans le but d'ajuster la réponse du réseau à l'expérience et aux exemples, mais aussi d'ajuster d'autres paramètres comme le taux d'apprentissage. Ces ajustements sont faits de sorte à minimiser l'erreur entre la sortie du réseau et le résultat désiré, en se basant sur le calcul d'une valeur de gradient [Chaoui, 2002]. Le gradient est la dérivée partielle de l'erreur par rapport aux poids synaptiques. Il sert à initialiser les poids d'un réseau et de minimiser l'erreur de sortie du réseau par rapport à l'ensemble de ses pondérations. Il permet de trouver les poids qui minimisent le nombre d'erreurs commises sur l'ensemble d'apprentissage.

La qualité d'apprentissage ne dépend pas uniquement de ces algorithmes. Elle dépend aussi de l'architecture du réseau de neurones et des fonctions d'activations choisies [Chaoui, 2002] .

III.4.4 Architecture de réseaux de neurones

Les réseaux de neurones comportent une partie conception, dont l'objectif est de permettre le choix de la meilleure architecture possible. L'architecture des réseaux de neurones artificiels joue un rôle important dans le processus d'apprentissage. Elle consiste à définir les différentes

connexions entre les neurones. On distingue trois structures de réseau, chaque architecture à sa propre organisation selon la topologie de connexion des neurones : Réseaux de neurones non bouclés, Réseaux de neurones récurrents, Réseaux de neurones convolutionnelles.

III.4.4.1 Réseaux de neurones non bouclés (réseaux de neurones artificiel)

Un réseau de neurones non bouclé (appelé aussi statique) est représenté graphiquement par un ensemble de neurones connectés entre eux. L'information circule des entrées vers les sorties sans retour en arrière ; c'est-à-dire à partir d'un neurone quelconque, en suivant les connexions, nous ne pouvons pas revenir au neurone de départ. Ce type de réseaux est utilisé pour effectuer des tâches d'approximation de fonction non linéaire, de la classification ou de la modélisation de processus statiques non linéaires, les réseaux de neurones non bouclé comporte deux types qui sont : perceptron et réseau à couches entièrement connectées. [Asradj, 2009] .

- **Perceptron**

Le perceptron simple est le premier réseau de neurones fonctionnel muni d'une méthode d'apprentissage. Il est linéaire et monocouche. Dans sa première conception, illustrée par la figure III.8, il est composé d'une couche de neurones d'entrée appelée rétine, qui reçoit les informations décrivant l'objet à analyser ou à reconnaître.

Tous les neurones de la rétine sont reliés par des liens pondérés à une couche de sortie (cellules de décision) [Dendani, 2010].

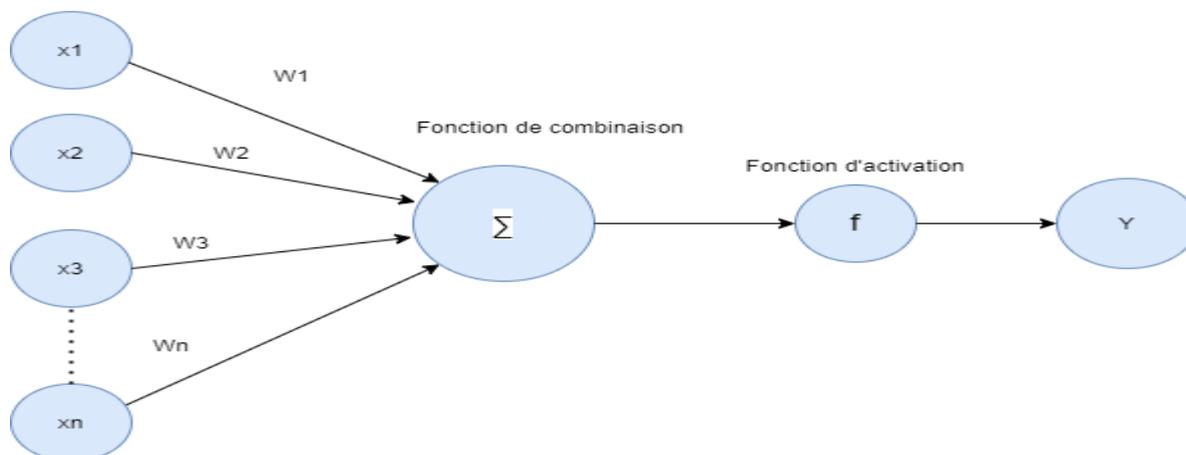


Figure III.8 la représentation de schéma d'un perceptron.

- **Réseau à couches entièrement connectées**

C'est le réseau de neurones statique le plus utilisé. Les neurones sont arrangés par couche. Les neurones de la première couche reçoivent des entrées, ils calculent leurs sorties qui sont transmises aux neurones de la seconde couche qui calculent eux même leurs sorties et ainsi de suite de couche en couche jusqu'à celle de sortie.

Chaque neurone dans la couche cachée est connecté à tous les neurones de la couche précédente et de la couche suivante, il n'y a pas de connexions entre les neurones d'une même couche. [Dendani, 2010].

La figure III.9 illustre l'exemple d'un réseau contenant trois entrées, deux couches cachées et une couche de sortie. La couche d'entrée représente toujours une couche virtuelle associée aux entrées du réseau, elle ne contient aucun neurone tandis que les couches suivantes représentent des couches effectives de neurones [Asradj, 2009] .

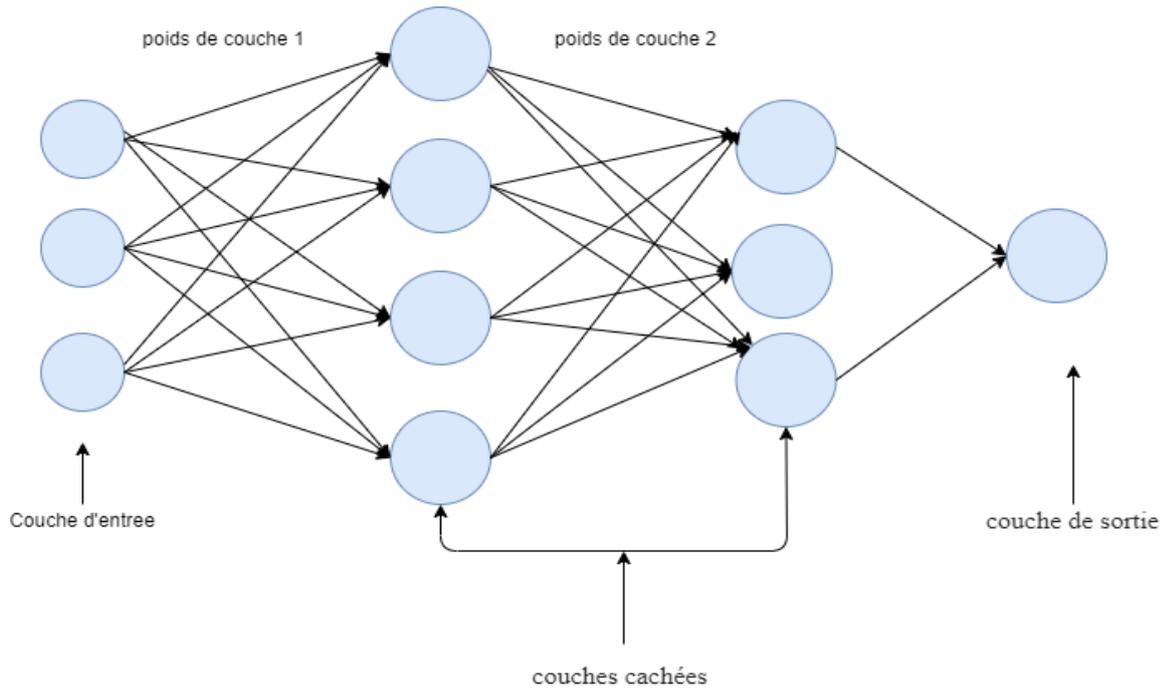


Figure III.9 Réseau de neurones à couches entièrement connectées

III.4.4.2 Réseaux récurrents

Les réseaux récurrents (Recurrent Neural Network) se distinguent des réseaux non bouclés par la connexion des sorties de neurones avec leurs entrées. La sortie d'un neurone peut être connectée avec l'entrée du même neurone ou avec celles des autres neurones.

L'importance de ces réseaux est qu'ils permettent d'apprendre la dynamique de système, c'est-à-dire qu'ils peuvent imiter le comportement temporel en insérant des délais dans les boucles, reliant l'entrée à la sortie du réseau ou dans des couches internes. [Chaoui, 2002] .

Ces réseaux sont utilisés pour effectuer des tâches de modélisation des systèmes dynamiques, de commande de processus ou de filtrage [Marzouka, 2009] .

La figure III.10 illustre un réseau récurrent dont les sorties sont bouclées avec les entrées de tous les neurones

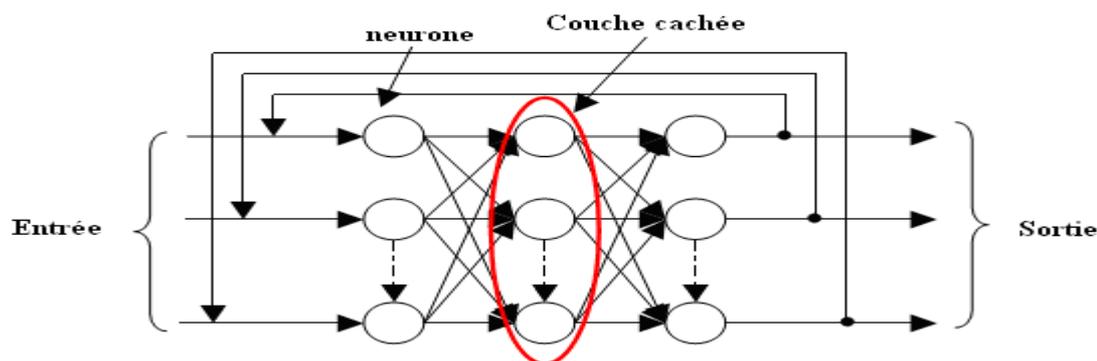


Figure III.10 Architecture d'un réseau récurrent.

- **LSTM (Long Short Term Memory)**

L'architecture LSTM consiste en un ensemble de sous-réseaux récurrents particuliers (appelés blocs de mémoire) situés au niveau de la couche cachée, chacun des sous-réseaux contient une ou plusieurs cellules de mémoire. Ces blocs de mémoire spéciaux permettent le maintien d'information en mémoire pour de longues périodes de temps. LSTM est proposé par Hochrieter et Shamid Huber, il est apparu comme un modèle efficace pour plusieurs problèmes d'apprentissage qui sont liés aux données séquentielles.

La figure III.11 illustre une cellule LSTM, il est composé d'une mémoire et de trois portes : la porte d'oubli, la porte d'entrée et la porte de sortie.

La porte d'oubli (forget gate) contrôle quelle est la partie de la cellule précédente qui sera oubliée et permet aussi de réinitialiser l'état de la mémoire au cours de la séquence. La porte d'entrée (input gate) doit choisir les informations pertinentes qui seront transmises à la mémoire.

La sortie (output gate) contrôle quelle partie de l'état de la cellule sera exposée en tant qu'état caché [Schmidhuber and Hochreiter, 1997].

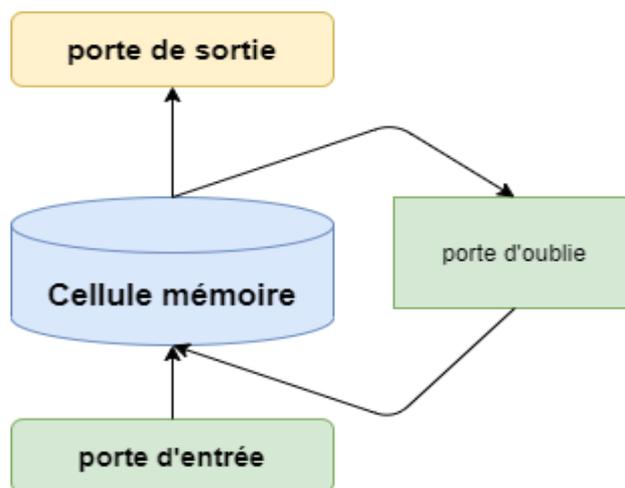


Figure III.11 Architecture d'un bloc LSTM.

- **GRU**

GRU (Gated Recurrent Unit) est un type de réseau récurrent. Qui a le même principe que le LSTM, mais son architecture est différente. Il est constitué que de deux portes : une porte de réinitialisation et une porte de mise-à-jour.

La porte de réinitialisation est une fusion de portes d'entrée et d'oubli. Elle détermine comment combiner la nouvelle entrée avec la mémoire précédente. La porte de mise à jour définit la quantité de mémoire précédente à conserver. L'intérêt de GRU par rapport au LSTM est d'être plus léger en termes de calculs pour une performance similaire [Bengio et al., 2014].

La figure III.12 illustre son architecture.

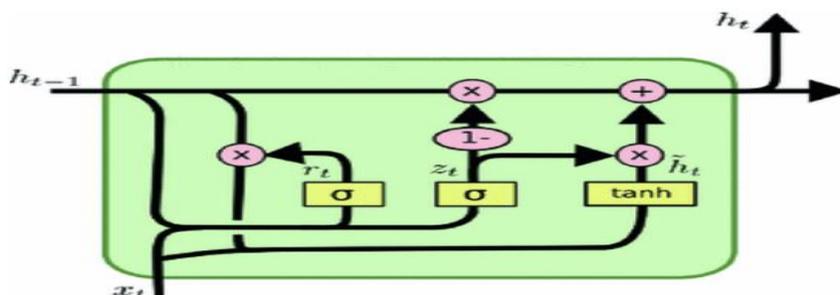


Figure III.12 Architecture de Réseaux GRU

Où :

$$z_t = \sigma(w_z \cdot [h_{t-1}, x_t])$$

$$r_t = \sigma(w_r \cdot [h_{t-1}, x_t])$$

$$\tilde{h}_t = \tanh(w_r \cdot [r_t * h_{t-1}, x_t]) \quad (3.5)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

III.4.4.3 Réseaux convolutionnels

Les réseaux de neurones convolutionnels (ou réseau de neurones à convolution, ou CNN ou ConvNet) sont des réseaux de neurones artificiels acycliques (non bouclés) dans lesquels la connexion entre les neurones est inspirée par le cortex visuel des animaux. Ce modèle de réseaux est proche d'un perceptron multicouche. Il est composé de neurones organisés en couche. Ces neurones ont des poids ou des paramètres, des biais et une fonction d'activation.

La principale différence est que les ConvNet sont réputés pour leur robustesse aux faibles variations d'entrée et le faible taux de prétraitement nécessaires à leur fonctionnement et ne requièrent aucun choix d'extracteur de caractéristiques spécifiques.

Ce modèle s'est révélé très efficace dans divers domaines tels que la reconnaissance d'images et de parole ainsi que dans la classification de textes.

Les CNN sont constitués essentiellement de deux types de neurones. Les neurones dits simples qui détectent les caractéristiques liées à la forme et les neurones dits complexes qui sont sensibles à la forme entière.

Dans les réseaux CNN chaque neurone prend des entrées, effectue une convolution (que nous présentons plus loin dans cette section) et il choisit éventuellement une fonction d'activation (non-linéarité).

Un réseau CNN possède en général plusieurs couches de convolutions et de pooling ainsi que des couches de correction et une couche entièrement connectée les unes à la suite des autres

avec des dimensions qui augmentent et se réduisent au fur et à mesure de l'avancée dans le réseau.

L'architecture CNN illustrées par la figure III.13. Elle contient des couches de convolution, de pooling, de correction et une couche entièrement connectée [Mokri, 2017].

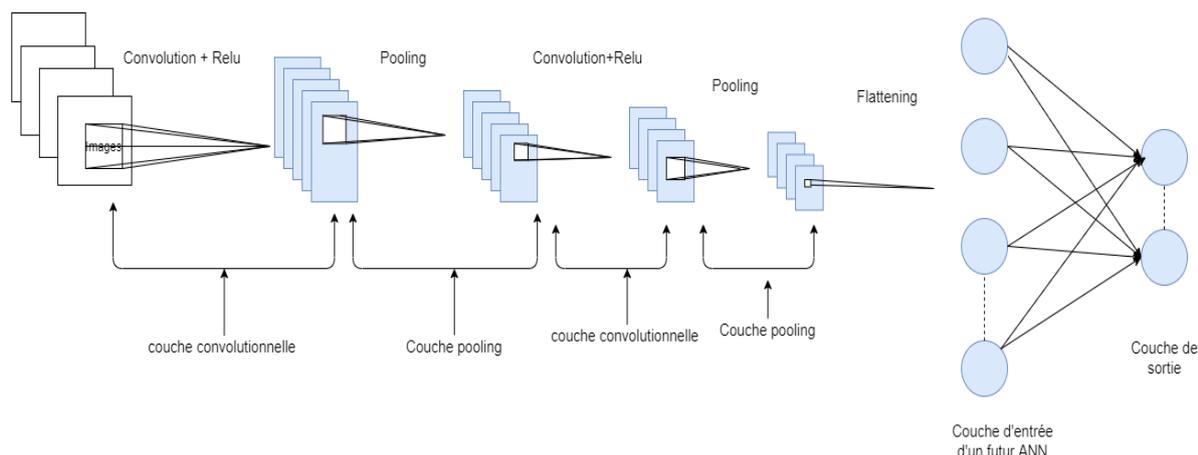


Figure III.13 Architecture d'un Réseaux Convolutionnels

En effet, avant de connecter à la couche entièrement connectée, les données qui sont en deux dimensions sont mises à plat et concaténées en un vecteur de caractéristiques, appelé code CNN.

- **Couche de convolution**

Dans la couche de convolution, au lieu de faire un produit scalaire entre les entrées et les poids de chaque neurone, on applique un produit de convolution.

Ce produit de convolution sert à extraire des caractères spécifiques des données en entrée.

Une donnée est donc passée à travers une succession de filtres, ou noyaux de convolution, créant de nouvelles données appelées cartes de convolutions. Le produit de convolution est la moyenne pondérée de l'ensemble d'entrée qui est défini par :

$$g(x, y) = \sum_{i,j}^{h_x, h_y} h(i, j) f(x + i, y + j) \quad (3.6)$$

Où f c'est l'entrée à traiter et h c'est le filtre à appliquer avec taille (h_x, h_y) .

Trois paramètres permettent de dimensionner le volume de la couche de convolution : la profondeur, le pas et la marge [Mokri, 2017].

Profondeur de la couche : c'est le nombre de noyaux de convolution qui correspond au nombre de neurones associés à un même champ récepteur (surface de traitement).

Le pas : contrôle le chevauchement des champs récepteurs. Plus le pas est petit, plus les champs récepteurs se chevauchent et plus le volume de sortie sera grand.

La marge : permet de contrôler la dimension spatiale du volume de sortie. En particulier, il est parfois préférable de conserver la même surface que celle du volume d'entrée [Mokri, 2017].

- **Couche de pooling**

Après chaque couche de convolution, il peut y avoir une couche de pooling (couche d'agrégation en français). Elle est une forme de sous-échantillonnage de l'entrée. Elle a un potentiel d'améliorer les résultats d'un CNN pour certaines applications et elle est souvent utilisée pour la classification de textes. Il est donc fréquent d'insérer périodiquement une couche de pooling entre deux couches convolutives successives d'une architecture CNN [Mokri, 2017].

L'étape de pooling consiste en une sélection de valeur (max, min, moyenne...) selon des fenêtres coulissantes à la matrice issue de la convolution (le résultat de produit de convolution). Il se traduit mathématiquement par la fonction maximum de plusieurs noyaux de convolution.

- **Couche de correction (Relu)**

Elle permet d'améliorer l'efficacité du traitement en intercalant entre les couches de traitement une couche qui va opérer une fonction mathématique (fonction d'activation Relu) sur les signaux de sortie. Cette fonction force les neurones à retourner des valeurs positives [Mokri, 2017].

- **Couche entièrement connectée (FC)**

Après plusieurs couches de convolution et de pooling, le raisonnement de haut niveau dans le réseau neuronal se fait via des couches entièrement connectées. Les neurones dans une couche entièrement connectée ont des connexions vers toutes les sorties de la couche précédente [Mokri, 2017].

III.4.5 Algorithmes d'apprentissage des réseaux de neurones

Toutes les informations que les réseaux de neurones contiennent, se résident dans les poids synaptiques. Donc l'algorithme d'apprentissage est la méthode mathématique qui va mettre à jour à chaque fois les poids afin de converger vers une meilleure solution qui permettra au réseau d'accomplir les tâches souhaitées. Pour cela nous allons présenter de ce qui suit quelques algorithmes d'apprentissage des réseaux de neurones :

III.4.5.1 Rétro propagation du gradient

La rétro propagation du gradient (Back propagation en anglais) est une méthode adaptée au réseau à couches entièrement connectées. Elle permet de calculer le gradient de l'erreur pour chaque neurone du réseau, de la dernière couche vers la première.

Le principe de la rétro propagation peut être décrit en trois étapes fondamentales :

- Acheminement de l'information à travers le réseau.
- Rétro propagation des sensibilités de réseau (propager l'erreur) et calcul du gradient.
- Ajustement des paramètres par la règle du gradient approximé [Ammar, 2007].

L'algorithme du gradient a pour but de converger de manière itérative vers une configuration optimisée des poids synaptiques qui permet d'adapter les poids des connexions du réseau de manière à minimiser la somme des erreurs sur tous les neurones de sortie afin d'obtenir une sortie voulue lors de l'entraînement. Cet état peut être un minimum local de la fonction à optimiser et idéalement, un minimum global de cette fonction (dite fonction de coût ou fonction d'erreur). Elle sert à calculer la moyenne des erreurs obtenues pour chacun des exemples de la base d'apprentissage. Cette moyenne calculée est dite minimale lorsque les poids et les biais sont ajustés).

La fonction de coût est non linéaire au regard des poids synaptiques. Elle dispose également d'une borne inférieure et moyennant quelques précautions lors de l'apprentissage, les procédures d'optimisation finissent par aboutir à une configuration stable au sein du réseau de neurones [Bendaoud, 2014].

III.4.5.2 Adagrad

AdaGrad est un algorithme qui vise à résoudre le problème de l'adaptation du taux d'apprentissage aux différents paramètres à optimiser. Il s'ajuste automatiquement en fonction de l'éparsité des paramètres (poids du réseau).

C'est un algorithme d'optimisation qui ne fait qu'adopter le taux d'apprentissage aux paramètres (poids du réseau), effectuant de plus grandes mises à jour pour les caractéristiques moins fréquentes et de plus petites mises à jour pour les caractéristiques plus fréquentes [Moualek, 2017]

III.4.5.3 Adam

Adam (Adaptive Moment Estimation) est l'une des méthodes d'optimisation basée sur les gradients. Son principe est le même que pour Adagrad. Elle est simple à mettre en œuvre et efficace sur le plan informatique. En outre, elle requiert également peu de mémoire, est invariante au redimensionnement diagonal des gradients. Ce type de méthode convient bien aux problèmes importants en termes de données ou de paramètres [Lie Ba and Kingma, 2015].

III.4.5.4 Nastrov-Adam

Nesterov-Adam (NAdam) est une méthode de descente de gradient très adaptée à la classification. Elle intègre la dynamique de la méthode de Nesterov momentum, elle accélère le processus de calcul du gradient dans la méthode d'optimisation. Elle utilise une combinaison de vitesse de modification des paramètres et de taux d'apprentissage adaptatif pour améliorer les calculs de gradient. Cela permet de fournir une méthode convergente efficace et rapide pour la descente de gradient [Zhou and Rossetto, 2019].

III.5 Apprentissage automatique et classification de textes

La classification d'une grande masse de documents semblerait très coûteuse en termes de temps, budget et personnel. Pour cela les experts de recherche se sont basés sur la classification automatique. Le système de classification de textes inclut un modèle de prédiction, qui a comme entrée un texte et en sortie une ou plusieurs étiquettes, en passant par quatre étapes principales illustrées dans la figure III.14.

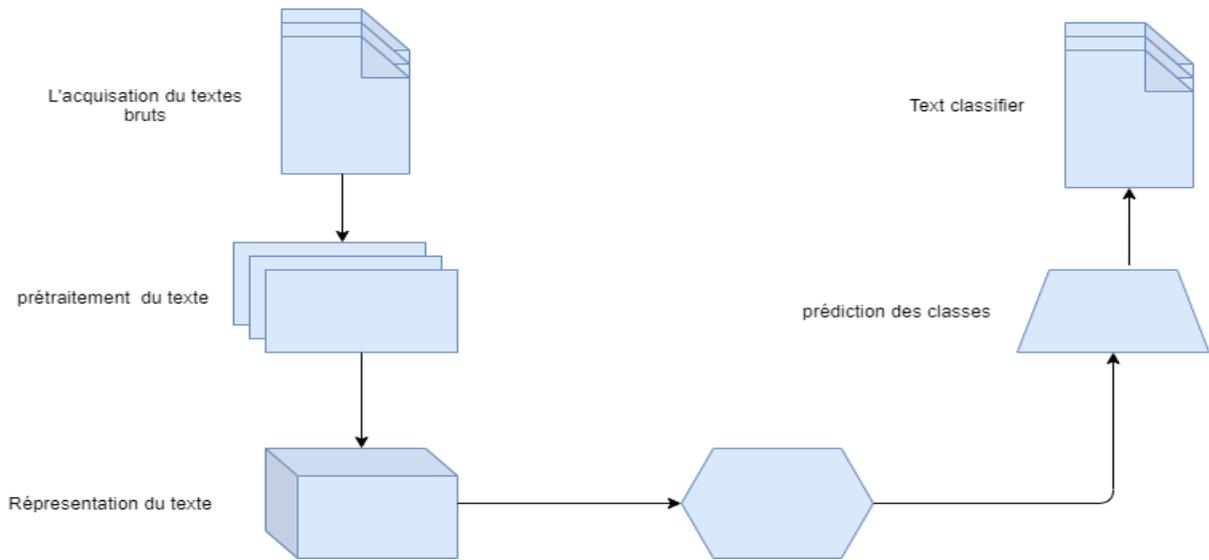


Figure III.14 Processus de classification de textes.

III.5.1 L'acquisition de textes brutes

Cette étape consiste à récupérer du texte brut. Dans notre cas nous avons un jeu de données déjà existants (documents).

III.5.2 Prétraitement de textes

L'étape de prétraitement consiste à formaliser les textes afin qu'ils soient compréhensibles par la machine et utilisables par les algorithmes d'apprentissage. L'objectif de cette phase est de collecter les caractéristiques (descripteurs) qui correspondent à un mot, un groupe de mots ou encore un concept, afin de les représenter sous forme de vecteur définissant le poids de chaque caractéristique.

Différentes techniques d'extraction des caractéristiques du texte sont disponibles. Elles sont liées au mode de représentation de ce texte

III.5.2.1 Nettoyage de mots

Cette technique permet de nettoyer le texte en éliminons les ponctuations. Ensuite découper les séquences de caractères en fonction de la présence ou l'absence de caractères de séparation (de type espace, tabulation ou retour chariot), puis supprimer les mots les plus fréquents. Eventuellement, unifier les écritures en lettres majuscules ou en lettres minuscules avant ou après les opérations déjà effectuées.

III.5.2.2 Suppression des mots vides

Dans cette étape, il faut éliminer les mots non significatifs qui n'apportent pas de valeur informative pour la compréhension du sens d'un texte tels que les articles, les conjonctions de coordination, les verbes auxiliaires...etc.

III.5.2.3 La tokenisation

Dite aussi segmentation du texte, son objectif est d'identifier les mots (tokens) en reconnaissant tout ce qui représente des séparateurs, des caractères spéciaux, des chiffres, les ponctuations, etc.

Par exemple la tokenisation en mots d'une phrase "je suis étudiant du département informatique " après nettoyage, donne les tokens suivants : "je", "suis" , "étudiant", "département ", "informatique ".

Elle permet aussi la transformation de l'ensemble des tokens en vecteur.

III.5.3 Représentation de textes

Après avoir vu les différentes techniques de prétraitements, cette étape consiste à extraire l'information du texte pour le traitement ultérieur par des modèles d'apprentissage profond.

Elle permet de présenter le texte comme un vecteur numérique interprétable par les méthodes de classification.

Pour la représentation des documents textuels, plusieurs méthodes sont utilisées, nous présentons quelques-unes dans ce qui suit.

III.5.3.1 Représentation en sac de mots (Bag of Words)

C'est la représentation la plus couramment utilisée dans laquelle les textes sont transformés simplement en vecteurs dont chaque composante représente un mot.

Cette représentation vise à représenter le texte avec une liste de mots simples contenus dans ce dernier.

Suivant les différentes techniques de prétraitement, certains mots comme les mots vides ne font pas partis du sac.

A chaque mot est associée une mesure, cette mesure peut être une valeur binaire (positive = 1 ; négative = 0), une fréquence d'apparition dans le document ou encore le nombre d'occurrences dans le texte [Fusiello et al., 2009].

Avec cette approche, les documents sont représentés par des vecteurs de dimension égale à la taille du vocabulaire , qui est en générale assez grand.

III.5.3.2 Représentation avec les n-grammes

De nombreux travaux ont montré l'efficacité des n-grammes comme méthode de représentation des textes pour leur classification : recherche d'une partition en groupes homogènes, ou pour leur catégorisation : attribution d'un texte à une, ou plusieurs, catégorie(s) parmi une liste prédéterminée. Un n-gramme est une séquence de n caractères consécutifs. Pour un document quelconque, l'ensemble des n-grammes (en général n prend les valeurs 2 ou 3) qu'on peut générer est le résultat qu'on obtient en déplaçant une fenêtre de n cases sur le corps de texte [Trenkle and Cavnar, 1994] .

III.5.3.3 Représentation par prolongement de mots (Word Embedding)

La représentation par plongement de mots ou encore dense des mots sont les représentations les plus utilisées actuellement dans les dernières méthodes de traitement du langage.

Les *word embeddings* sont représentés par des vecteurs denses, un vecteur représentant la projection du mot dans un espace vectoriel continu. Ces derniers sont conçus pour capturer ce que nous appelons des similarités entre les mots du vocabulaire :

Les mots qui apparaissent dans des contextes similaires seront proches les uns des autres dans l'espace de projection [Vedantam et al., 2016].

Le principe est de regrouper les mots ayant les mêmes propriétés sémantique ou syntaxique.

III.5.4 Choix de classifieur

Cette dernière étape consiste à faire le traitement de textes, que ce soit par l'algorithme d'apprentissage ou celui de classification.

La catégorisation de textes comporte un choix de technique d'apprentissage (ou classifieur) disponibles. Il existe plusieurs méthodes d'apprentissage et parmi les plus souvent utilisées figurent les classifieurs naïve bayes, les machines à vecteurs de support et les classifieurs à base réseaux de neurones.

III.5.4.1 Les classifieurs naïf bayes

Les classifieurs naïf Bayésien est une méthode appartenant à la catégorie des méthodes de classification Bayésienne probabiliste fondée sur le théorème de Bayes [Lewis, 1998]. L'objectif de cette méthode est le calcul des probabilités à posteriori qu'un document appartient à chacune des catégories de l'analyse. La classe ayant la plus forte probabilité est sélectionnée comme label du document.

III.5.4.2 Les machines à vecteurs de support

Les machines à vecteurs de support sont une méthode d'apprentissage supervisée qui est très utilisée pour la classification de textes. L'objectif du SVM est la recherche d'un hyperplan de marge maximale séparant les données d'apprentissage en deux classes d'analyse [Gammerman et al., 1998].

III.5.4.3 Les classifieurs à base de réseaux de neurones

L'apprentissage neuronal est appliqué avec succès dans les tâches de classification de textes. Le réseau de neurones Est un réseau d'unités construit à partir des documents d'apprentissage, où les unités d'entrées représentent les pondérations des termes, les unités de sortie représentent les catégories, et les arcs reliant les unités représentent les relations d'indépendances [Barigou, 2013].

L'activation des unités d'entrées est propagée à travers le réseau et la valeur de l'unité de sortie détermine la classe du document.

III.5.5 Corpus de Textes

Afin de tester l'ensemble de ces algorithmes et techniques, il est nécessaire de disposer de matières premières : des textes labellisés. Obtenir un corpus de textes labellisés peut parfois être une étape compliquée car ces derniers sont coûteux à produire.

En effet, chaque texte doit être labellisé manuellement pour permettre à la machine de retrouver le modèle. De plus, afin de comparer les résultats des différentes méthodes et techniques, il est nécessaire de disposer de corpus libres [RISCH, 2017] .

III.5.6 Difficultés particulière de la classification de textes

Beaucoup de difficultés peuvent s'opposer au processus de classification de textes. Des problèmes connus dans la discipline liée à l'apprentissage automatique comme la subjectivité de la décision prise par les experts, ainsi que des problèmes liés à la nature des données traitées à savoir des données textuelles comme la polysémie, la redondance, les variations morphologiques ou même L'homographie, etc.

Nous citons les trois principales Dans ce qui suit.

III.5.6.1 Polysémie

Un mot possède, dans différents cas, plus d'un sens et plusieurs définitions lui sont associées. Par conséquent, à cause de la polysémie, les mots seuls sont parfois de mauvais descripteurs ; exemple le mot livre peut désigner une unité monétaire, ou un bouquin [Rimouche and Hachemi, 2015] .

III.5.6.2 Les mots composés

La non prise en charge des mots composés comme : Arc-en-ciel, peut-être, sauve-qui-peut, etc, dont le nombre est très important dans toutes les langues.

Traiter le mot Arc-en-ciel par exemple en étant 3 termes séparés réduit considérablement la performance d'un système de classification néanmoins l'utilisation de la technique des n-grammes pour le codage des textes atténue considérablement ce problème des mots composés [Azizi , 2018].

III.5.6.3 Redondance

La redondance et la synonymie permettent d'exprimer le même concept par des expressions différentes. Cette difficulté est liée à la nature des documents traités exprimés en langage naturel contrairement aux données numériques [Azizi , 2018].

III.5.6.4 Subjectivité de la décision

Parmi les problèmes classiques usuels dans le domaine de l'apprentissage automatique c'est la subjectivité de la décision prise par les experts qui décident de la classe à laquelle le texte va être attribué [Azizi , 2018].

III.6 Conclusion

Dans ce chapitre nous avons présenté les généralités et quelques concepts des réseaux de neurones, ses architectures, ses fonctions d'activation, ainsi que quelques algorithmes d'apprentissage. Enfin nous avons présenté la classification automatique de textes.

Ces algorithmes peuvent être utilisés dans divers domaines d'application telle que la classification de texte dans la RI.

Le prochain chapitre porte sur notre approche d'inférence du profil utilisateur en se basant sur les algorithmes de réseaux de neurones.

**Chapitre 04 : Définition d'un profil
utilisateur en se basant sur les réseaux de
neurones.**

IV.1 Introduction

La modélisation des profils utilisateurs est au cœur d'un système de recherche d'information personnalisé pour mieux répondre à ses besoins en information.

Notre contribution présentée dans ce chapitre porte sur la définition d'un profil utilisateur en se basant sur les réseaux de neurones.

Nous avons proposé différents modèles à base de ces réseaux pour deux phases complémentaires afin d'inférer un profil utilisateur. Enfin, nous terminons par une conclusion.

IV.2 Motivations et objectifs

La RI personnalisée s'intéresse à intégrer le profil utilisateur dans l'une des phases du processus de recherche d'information pour mieux répondre à ses besoins. Différents travaux s'intéressent sur la manière de construction de ce profil.

Notre approche consiste à définir un profil utilisateur en se basant sur les réseaux de neurones. Nous proposons de définir ce profil par son centre d'intérêt.

Nous allons utiliser les documents jugés implicitement pertinent par un utilisateur (clic + durée de lecture) pour extraire automatique le domaine de chaque document. Ce domaine peut correspondre au centre d'intérêt d'un utilisateur.

L'idée est de pouvoir exploiter au mieux le contenu du document afin de définir son domaine en utilisant les réseaux de neurones. Ces derniers ont montré leurs efficacités dans de nombreux domaines d'applications tels que le traitement d'image, la catégorisation de textes et le diagnostic médicale et cela grâce à leur capacité de classification et de généralisation.

Dans notre cas, nos modèles de réseaux de neurones déployés, en s'appuyant sur différentes architectures, sont utilisés comme méthode de classification de texte d'un document selon une catégorie.

Chapitre 4 : Définition d'un profil utilisateur en se basant sur les réseaux de neurones

Pour que le profil soit plus spécifique et plus riche, nous allons exploiter ce centre d'intérêt avec d'autres caractéristiques de l'utilisateur, afin de prédire des centres d'intérêts plus spécifiques.

IV.3 Architectures des modèles proposés

Notre approche est divisée en deux phases :

La première phase consiste à déduire le domaine du document jugé implicitement pertinent par un utilisateur qui correspond à son centre d'intérêt, pour cela nous avons créé trois modèles avec différentes architectures.

La deuxième phase consiste à utiliser le domaine prédit par le meilleur modèle de la première phase avec d'autres caractéristiques de l'utilisateur afin de prédire d'autres centres d'intérêts.

Nous présentons dans ce qui suit les architectures de chaque phase.

IV.3.1 Architecture des modèles de la première phase

La première phase permet de déduire les domaines des documents. Le domaine prédit correspond au centre d'intérêt de l'utilisateur.

Pour la modélisation de cette phase nous avons proposé trois différentes architectures. Chacune d'elles est caractérisée par :

- Le type de réseaux de neurones utilisée à savoir : ANN, CNN et RNN.
- Le nombre de couche cachée.
- Nombre d'unités de chaque couche.

IV.3.1.1 Premier modèle

Nous proposons pour ce premier modèle une architecture ANN décrite dans la figure IV.1.

Chapitre 4 : Définition d'un profil utilisateur en se basant sur les réseaux de neurones

Elle est composée d'une couche d'entrée, couche average pooling, deux couches entièrement connectées et une couche de sortie Chacune des couches précédentes a une fonction d'activation Relu ou SoftMax).

La première couche de notre réseau est une couche d'incorporation qui code le texte en entrée.

Elle représente une description de taille 200 en une séquence de vecteurs denses de dimension 64.

La taille des descriptions est fixée à 200. Cette valeur représente la taille de la plus longue description dans notre jeu de données.

La couche average pooling permet de prendre la moyenne du patch d'entrée.

Les deux autres couches cachées ont respectivement 130 neurones et 70 neurones où la fonction d'activation utilisée est Relu.

La dernière couche utilise la fonction d'activation softmax qui permet de calculer la distribution de probabilité des cinq classes. Le nombre de neurones est décroissant, cela permet d'avoir un passage progressif d'un nombre élevé de neurones en entrée, vers 5 neurones en sortie, pour faciliter l'entraînement et sans pour autant rendre le réseau trop complexe.

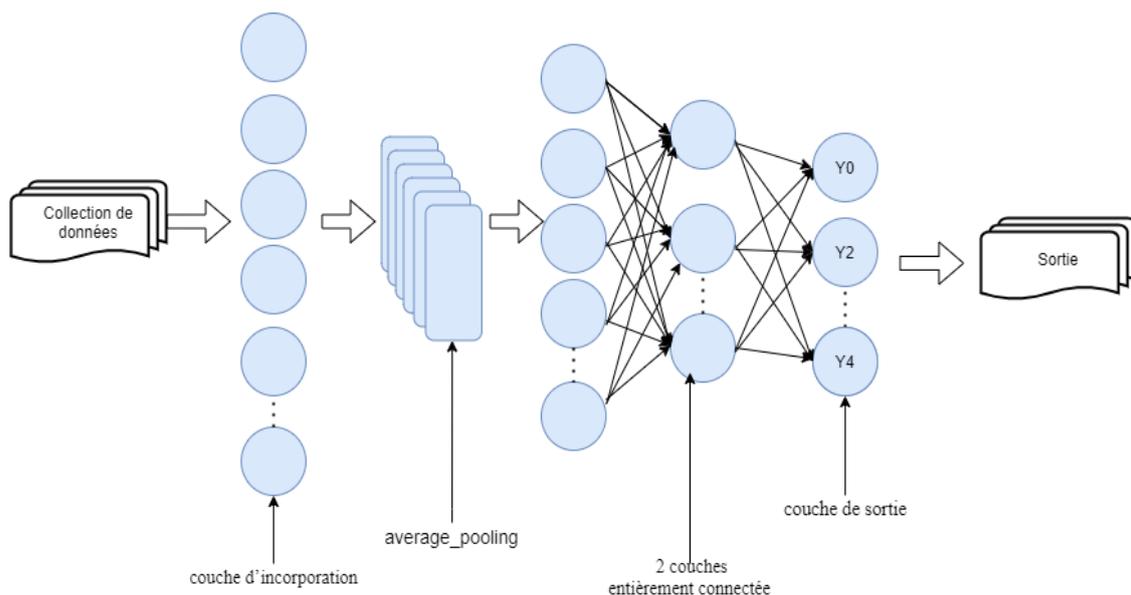


Figure IV.1 Architecture du premier modèle

IV.3.1.2 Deuxième modèle

Le deuxième modèle que nous présentons dans la figure IV.2 est une architecture simple d'un réseau LSTM.

La première couche de notre réseau est une couche d'incorporation qui code le texte en entrée qui représente une description de taille 200 en une séquence de vecteurs denses de dimension 64.

La couche suivante est une couche LSTM (bidirectionnel) qui transforme notre entrée en un seul vecteur contenant des informations sur la séquence entière. Elle utilise 128 neurones, le résultat obtenu est ensuite passé aux deux couches entièrement connectées de 128 et 64 neurones respectivement.

Le résultat de sortie est transmis vers la couche de sortie utilisant 5 neurones où la fonction d'activation est softmax.

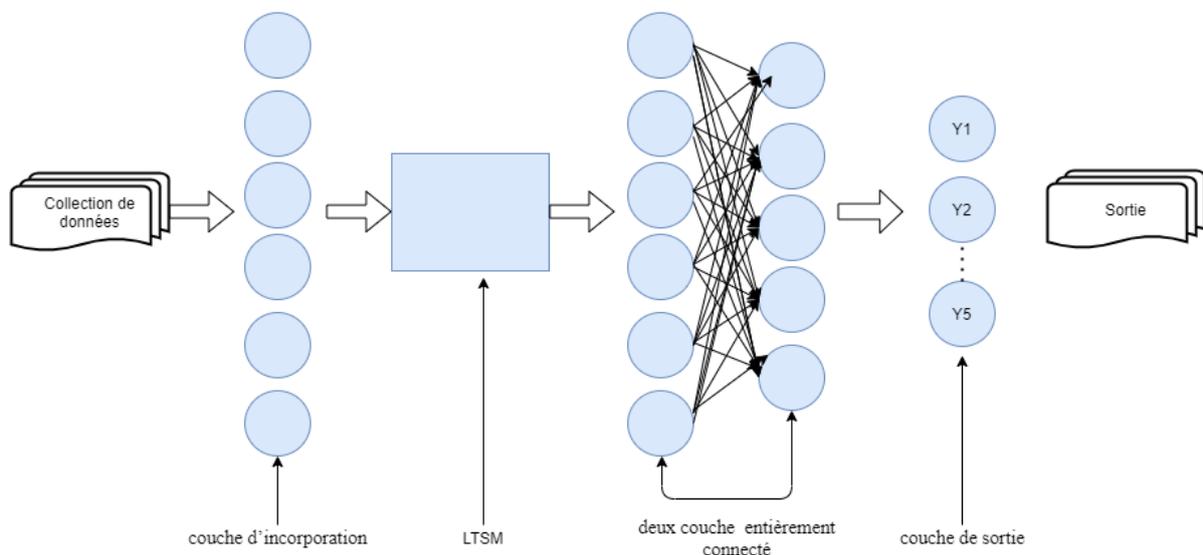


Figure IV.2 Architecture du deuxième modèle

IV.3.1.3 Troisième modèle

Le troisième modèle que nous présentons dans la figure IV.3 est une architecture de réseau convolutionnel (CNN). Elle est composée d'une couche d'incorporation, une couche de convolution, une couche de average pooling et une couche entièrement connectée, et la couche de sortie.

La première couche de notre réseau est une couche d'incorporation qui code le texte en entrée et qui représente une description de taille 200, en une séquence de vecteurs denses de dimension 64, la deuxième couche est la couche de convolution avec 128 neurones, utilisant une fenêtre de convolution de taille 5. Le résultat est ensuite passé à une couche de average pooling afin de compresser l'information en réduisant la taille de notre entrée.

Elle utilise 128 neurones et une fenêtre de pooling de taille 2, le résultat de la sortie sera transmis à une couche entièrement connectée de 64 neurones.

La dernière couche utilise la fonction d'activation softmax qui permet de calculer la distribution de probabilité des 5 classes.

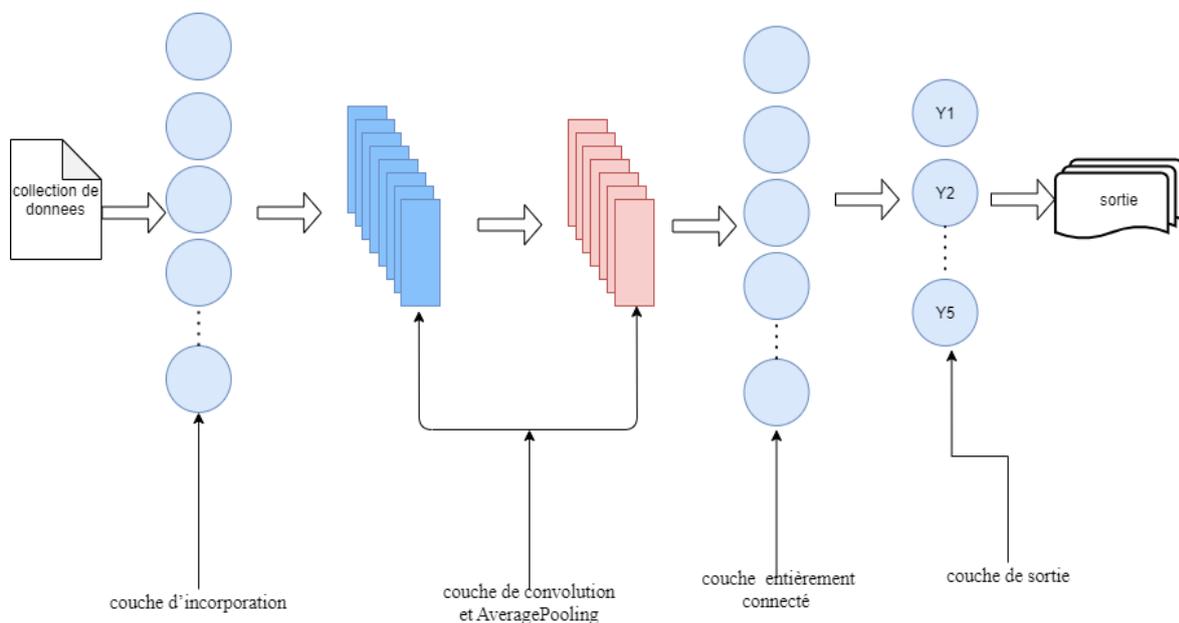


Figure IV.3 Architecture du troisième modèle

IV.3.2 Architectures des modèles de la deuxième phase

Cette deuxième phase permet d'exploiter le domaine prédit par le meilleur modèle de la première phase avec d'autres caractéristiques de l'utilisateur comme l'âge, genre, salaire, localisation, afin de prédire d'autres centres d'intérêts. Par exemple nous avons un centre d'intérêt plus générale comme sport nous allons déduire un sous centre d'intérêt par exemple football à partir des profils existants.

Nous présentons dans ce qui suit les deux différentes architectures de la deuxième phase.

Elles diffèrent selon le nombre d'unité de chaque couche.

IV.3.2.1 Premier modèle

Pour ce premier modèle nous proposons une architecture ANN décrite dans la figure IV.4.

Le réseau est constitué d'une couche d'entrée de six unités, deux couches cachées qui ont six unités avec une fonction d'activation relu .

Le résultat de sortie est transmis vers la couche de sortie utilisant 15 neurones où la fonction d'activation est softmax.

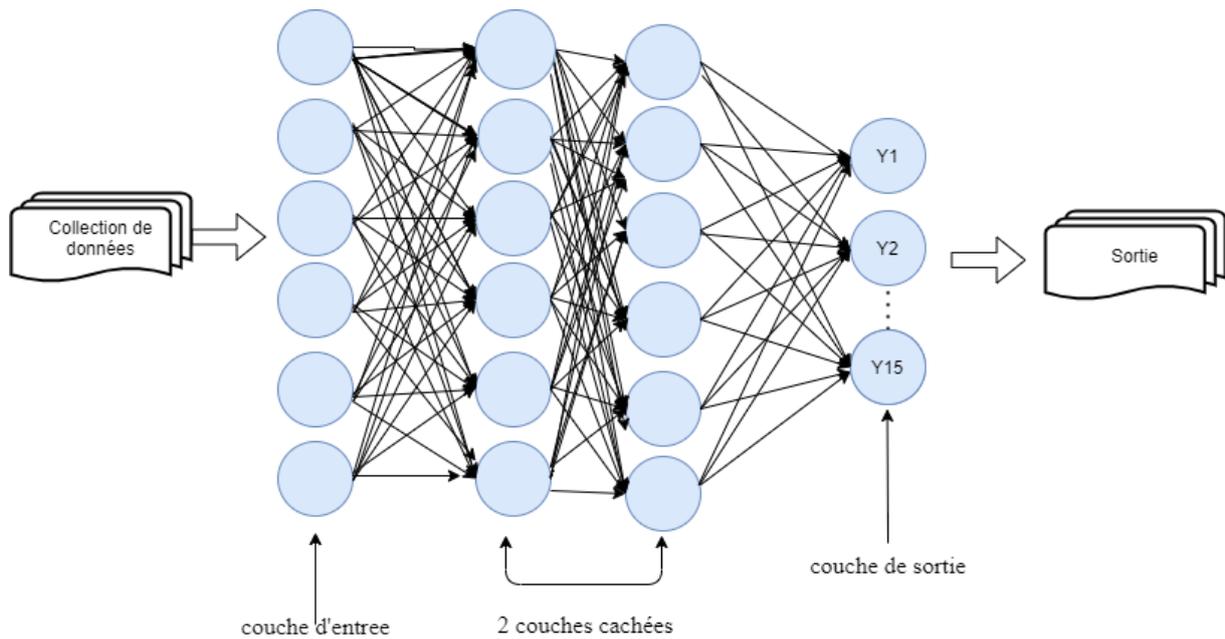


Figure IV.4 Architecture du premier modèle de la deuxième phase

IV.3.2.2 Deuxième modèle

Pour ce second modèle nous proposons une architecture ANN décrite dans la figure IV.5, le réseau est constitué d'une couche d'entrée de 50 neurones et deux couches cachées qui ont 40 et 20 neurones respectivement avec une fonction d'activation relu.

Le résultat de sortie est transmis vers la couche de sortie utilisant 15 neurones où la fonction d'activation est softmax.

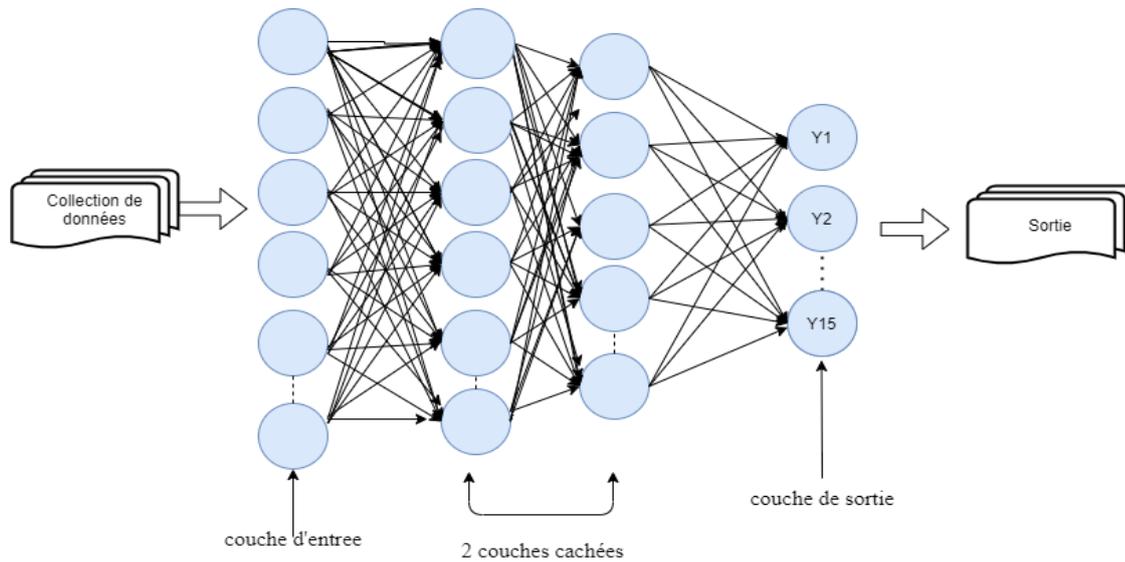


Figure IV.5 Architecture du deuxième modèle de la deuxième phase

IV.4 Conclusion

Dans ce chapitre, nous avons décrit les architectures des réseaux de neurones que nous avons proposé afin d'extraire les centres d'intérêts représentant le profil utilisateur, en se basant sur les documents jugés implicitement pertinents.

Le prochain chapitre décrit l'implémentation de ces architectures, les expérimentations réalisées sur les modèles implémentés ainsi que le prétraitement de jeux de données.

Chapitre 5 : Implémentation et expérimentations

V.1 Introduction

Dans ce chapitre, nous présentons les outils logiciels ainsi que les différentes bibliothèques que nous avons utilisées. Parmi elles Tensorflow et Keras qui sont les deux principales bibliothèques déployées pour la réalisation et l'apprentissage de nos modèles.

Nous présentons les Jeux de données et la construction des collections d'entraînement, de validation et de test, ainsi que leurs prétraitements et leurs représentations.

Par la suite, nous montrons les résultats obtenus pour chaque architecture proposée et effectuons une comparaison entre elles afin de définir la mieux adaptée à notre approche.

V.2 Présentation des outils

V.2.1 Logiciels

- **Anaconda :**

Est une distribution des langages de programmation Python et R appliqué au développement d'applications dédiées à la science des données et à l'apprentissage automatique.

- **Spyder (IDE) :**

Est un environnement de développement scientifique puissant écrit en Python. Il est conçu pour des scientifiques, des ingénieurs et des analystes de données.

Il présente une combinaison unique des fonctionnalités avancées d'édition, d'analyse, de débogage et de profilage d'un outil de développement complet avec l'exploration de données, l'exécution interactive, l'inspection approfondie et les superbes capacités de visualisation d'un package scientifique.

V.2.2 Bibliothèques logicielles

- **Tensorflow¹**

Est un framework de programmation pour le calcul numérique qui a été rendu Open Source par Google en Novembre 2015. C'est l'un des framework les plus utilisés pour l'apprentissage profond et donc les réseaux de neurones.

Son nom est notamment inspiré du fait que les opérations courantes sur des réseaux de neurones sont principalement faites via des tables de données multidimensionnelles, appelées Tenseurs (Tensor). Un tenseur à deux dimensions est l'équivalent d'une matrice.

- **Keras²**

Est une API de réseaux de neurones de haut niveau, écrite en Python et capable de fonctionner sur Tensorflow. Elle présente un ensemble d'abstractions de niveau supérieur et plus intuitif qui facilitent la configuration des réseaux neuronaux indépendamment de la bibliothèque informatique de backend.

- **Pandas³**

Est une bibliothèque écrite pour le langage de programmation Python permettant la manipulation et l'analyse des données. Elle propose en particulier des structures de données et des opérations de manipulation de tableaux numériques et de séries temporelles.

Pandas est un logiciel libre sous licence BSD, ce qui permet sa réutilisation avec peu de restrictions.

¹ <https://www.spyder-ide.org/>

² <https://keras.io/>

³ <https://pandas.pydata.org>

- **Numpy⁴**

Est la librairie fondamentale du calcul scientifique avec Python, destinée à manipuler des matrices ou tableaux multidimensionnels ainsi que des fonctions mathématiques opérant sur ces tableaux (array).

Les calculs avec Numpy sont particulièrement optimisés car les tableaux sont homogènes (ils ne contiennent que des valeurs d'un même type) et de taille fixée à la création

- **Csv-Python**

Le format CSV (Comma Separated Values) est le format d'importation et d'exportation utilisé par les feuilles de calcul et les bases de données. Csv est une bibliothèque qui implémente des classes pour lire et écrire des données tabulaires au format CSV.

Elle permet d'écrire les données dans le format Excel ou lire les données de fichier généré par Excel.

- **Os**

C'est une bibliothèque python qui fournit des fonctions permettant d'interagir avec le système d'exploitation, en particulier, avec le système de fichiers. Elle fait partie des modules utilitaires standard de Python. Elle fournit un moyen portable d'utiliser les fonctionnalités dépendantes du système d'exploitation .

Nous avons utilisé cette bibliothèque pour parcourir les répertoires contenant notre collections une fois celles-ci construite et afin de les utiliser pour l'entraînement et l'évaluation.

⁴ <https://www.numpy.org/>

- **Sys**

Cette bibliothèque fournit un certain nombre de fonctions et de variables pouvant être utilisées pour manipuler différentes parties de l'environnement d'exécution Python.

Nous avons utilisé cette bibliothèque pour manipuler les variables d'environnement (arg).

- **Matplotlib**

C'est une bibliothèque de traçage Python à deux dimensions qui produit des images de qualité. Elle peut être utilisée dans les scripts Python, les serveurs d'applications Web, etc. Matplotlib permet de générer des histogrammes, des graphiques à barres, des diagrammes d'erreur, des diagrammes de dispersion, etc, avec seulement quelques lignes de code (56).

Nous avons utilisé cette bibliothèque pour tracer des graphes afin de présenter le résultat de notre approche.

V.2.3 Configuration matérielle utilisée

La configuration matérielle utilisée pour l'entraînement et l'évaluation de nos modèles est la suivante :

- Processeur intel Core i7-5600U CPU contenant quatre coeurs à 2.60 GHZ chacun.
- Carte graphique intel HD Graphics 5500 2 GO.
- RAM d'une taille de 4 GO.
- Disque dur (SSD) d'une taille de 240 GO.
- Système d'exploitation Windows 10 Pro 64 bits.

V.3 Jeux de données utilisés

Pour la validation de notre approche, nous avons utilisée deux collections de données que nous présentons dans ce qui suit.

V.3.1 Première collection

Cette collection regroupe un ensemble de données public de la BBC composé de 2225 articles, organisés en cinq dossiers : affaires, divertissement, politique, sport et technologie.

Pour l'exploitation de cette collection, nous avons étiqueté chaque article par sa catégorie correspondante.

Le résultat est présenté dans la figure suivante.

1	category	text
2	tech	tv future in the hands of viewers with home theatre systems plasma high-definition tvs and digital video recorders moving into the li
3	business	worldcom boss left books alone former worldcom boss bernie ebbers who is accused of overseeing an \$11bn (Â£5.8bn) fraud neve
4	sport	tigers wary of farrell gamble leicester say they will not be rushed into making a bid for andy farrell should the great britain rugby lea
5	sport	yeading face newcastle in fa cup premiersip side newcastle united face a trip to ryman premier league leaders yeading in the fa cup
6	entertainmer	ocean s twelve raids box office ocean s twelve the crime caper sequel starring george clooney brad pitt and julia roberts has gone st
7	politics	howard hits back at mongrel jibe michael howard has said a claim by peter hain that the tory leader is acting like an attack mongrel s
8	politics	blair prepares to name poll date tony blair is likely to name 5 may as election day when parliament returns from its easter break the
9	sport	henman hopes ended in dubai third seed tim henman slumped to a straight sets defeat in his rain-interrupted dubai open quarter-fina
10	sport	wilkinson fit to face edinburgh england captain jonny wilkinson will make his long-awaited return from injury against edinburgh on sa
11	entertainmer	last star wars not for children the sixth and final star wars movie may not be suitable for young children film-maker george lucas has
12	entertainmer	berlin cheers for anti-nazi film a german movie about an anti-nazi resistance heroine has drawn loud applause at berlin film festival. s
13	business	virgin blue shares plummet 20% shares in australian budget airline virgin blue plunged 20% after it warned of a steep fall in full year p
14	business	crude oil prices back above \$50 cold weather across parts of the united states and much of europe has pushed us crude oil prices abo
15	politics	hague given up his pm ambition former conservative leader william Hague says he will not stand for the leadership again having give
16	sport	moya emotional after davis cup win carlos moya described spain s davis cup victory as the highlight of his career after he beat andy ro

Figure V.1 Structure de la première collection

Chapitre 5 : Implémentation et expérimentations

Nous avons divisé les données de notre collection en deux parties :

- **Données d'entraînement** : qui constituent 80% de notre jeu de données, à savoir 1780 document. Ces données sont utilisées pour générer le modèle d'apprentissage.
- **Données de test** : qui constituent les 20% restantes de notre jeu de données, à savoir 445 document. Le modèle d'apprentissage sera alors appliqué à ces données à des fins de tests.

V.3.2 La deuxième collection

Cette collection regroupe un ensemble de 10000 enregistrements que nous avons défini sous format CSV (Comma-separated-values). Ce format est de type textuel, ouvert représentant des données tabulaires sous forme de valeurs séparées par des virgules. Elles sont représentées sous formes de colonnes à savoir :

Utilisateur, id-utilisateur, domaine, âge, genre, localisation, salaire, centre d'intérêt.

La figure suivante présente les données de cette collection.

RowNumber	id_user	Surname	Geography	Gender	Age	field	Salary	center of interest
1	15634602	Hargrave	France	Female	42	tech	101348.88	computer scientist
2	15647311	Hill	Spain	Female	41	sport	112542.58	arbitrator
3	15619304	Onio	France	Female	42	sport	113931.57	arbitrator
4	15701354	Boni	France	Female	39	tech	93826.63	automation engineer
5	15737888	Mitchell	Spain	Female	43	Entraînement	79084.1	actor
6	15574012	Chu	Spain	Male	44	sport	149756.71	footballer
7	15592531	Bartlett	France	Male	50	Entraînement	10062.8	singer
8	15656148	Obinna	Germany	Female	29	sport	119346.88	arbitrator
9	15792365	He	France	Male	44	Entraînement	74940.5	actor
10	15592389	H?	France	Male	27	Entraînement	71725.73	animator
11	15767821	Bearce	France	Male	31	Entraînement	80181.12	actor
12	15737173	Andrews	Spain	Male	24	Entraînement	76390.01	actor
13	15632264	Kay	France	Female	34	Entraînement	26260.98	singer
14	15691483	Chin	France	Female	25	business	190857.79	business-developer
15	15600882	Scott	Spain	Female	35	Entraînement	65951.65	animator

Figure V.2 Structure de la deuxième collection

Nous avons divisé les données de notre collection en deux parties :

- **Données d'entraînement** : qui constituent 80% de notre jeu de données, à savoir 8000 profils. Ces données sont utilisées pour générer le modèle d'apprentissage.
- **Données de test** : qui constituent les 20% restantes de notre jeu de données, à savoir 2000 profils. Le modèle d'apprentissage sera alors appliqué à ces données à des fins de tests.

V.4 Mesures d'évaluation

Lorsqu'un modèle est développé, il est important d'être en mesure d'évaluer son fonctionnement et sa capacité à répondre aux objectifs qui lui ont été fixés.

Pour ce faire, nous avons utilisé deux mesures d'évaluation : la précision et la perte.

- **Précision (accuracy)**

La précision est la mesure la plus utilisée pour juger et évaluer la performance d'un modèle. Elle est définie par la formule suivante :

$$\text{Accuracy} = \frac{\text{Nombre de Prédictions correctes}}{\text{Nombre total de prédictions faites}} \quad (5.1)$$

- **Perte (loss)**

La perte indique dans quelle mesure un modèle donné se comporte bien après chaque itération d'optimisation. Idéalement, il doit y avoir une réduction des pertes après chaque ou plusieurs itérations.

V.5 Prétraitement et représentation

Dans cette section, nous avons présenté les différentes techniques que nous avons utilisées pour le prétraitement et la représentation des données des deux collections.

V.5.1 Première collection

Pour le prétraitement des données de la première collection, nous avons nettoyé les documents ainsi leurs contenus de tous les mots et caractères inutiles. Ensuite, nous avons appliqué une tokenisation à ces derniers.

Après avoir terminé la tâche de prétraitement, nous avons obtenu des documents textuels étiquetés.

Nous avons utilisé Glove pour la représentation des mots qui est une technique de word embedding. Glove signifie "vecteurs globaux pour la représentation de mots". C'est un dictionnaire qui permet d'associer à chaque mot un vecteur. Pour cela, il recueille les caractéristiques de cooccurrence de mots sous forme de matrice, pour construire des représentations compactées des documents.

La matrice de cooccurrences M est de dimension $V \times V$, avec V qui est le nombre de mots dans notre vocabulaire (la collection du document).

La figure suivante illustre les processus de traitement.

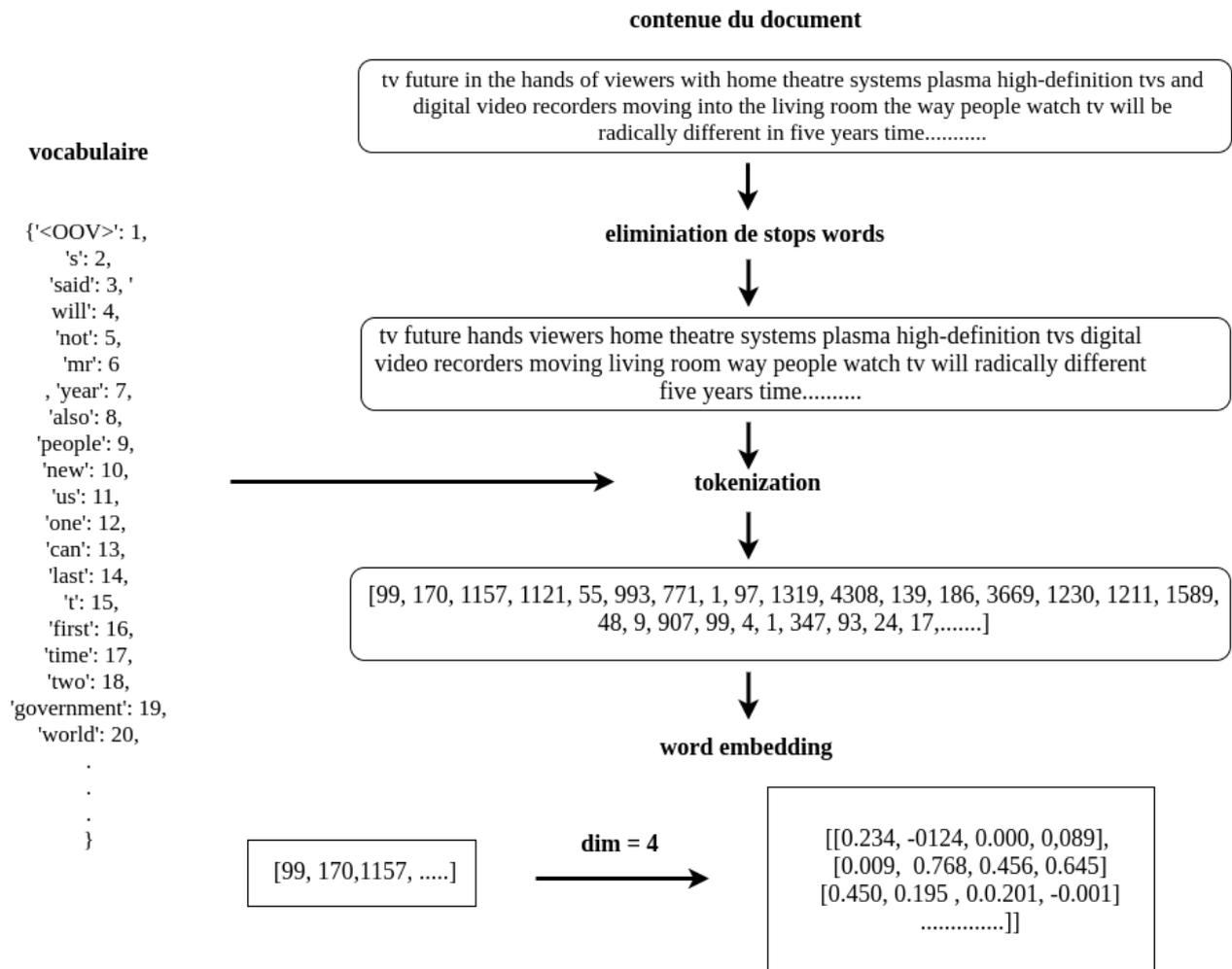


Figure V.3 Prétraitement et représentation de la première collection

V.5.2 Deuxième Collection

Pour le prétraitement et la représentation des données de la deuxième collection à savoir : âge, salaire, pays, genre et centre d'intérêt, nous les avons transformé sous forme numérique.

Ces valeurs seront représentées sous forme de vecteur qui sera prêts pour la phase d'entraînement.

La figure suivante explique en détaille les processus de traitement

Chapitre 5 : Implémentation et expérimentations

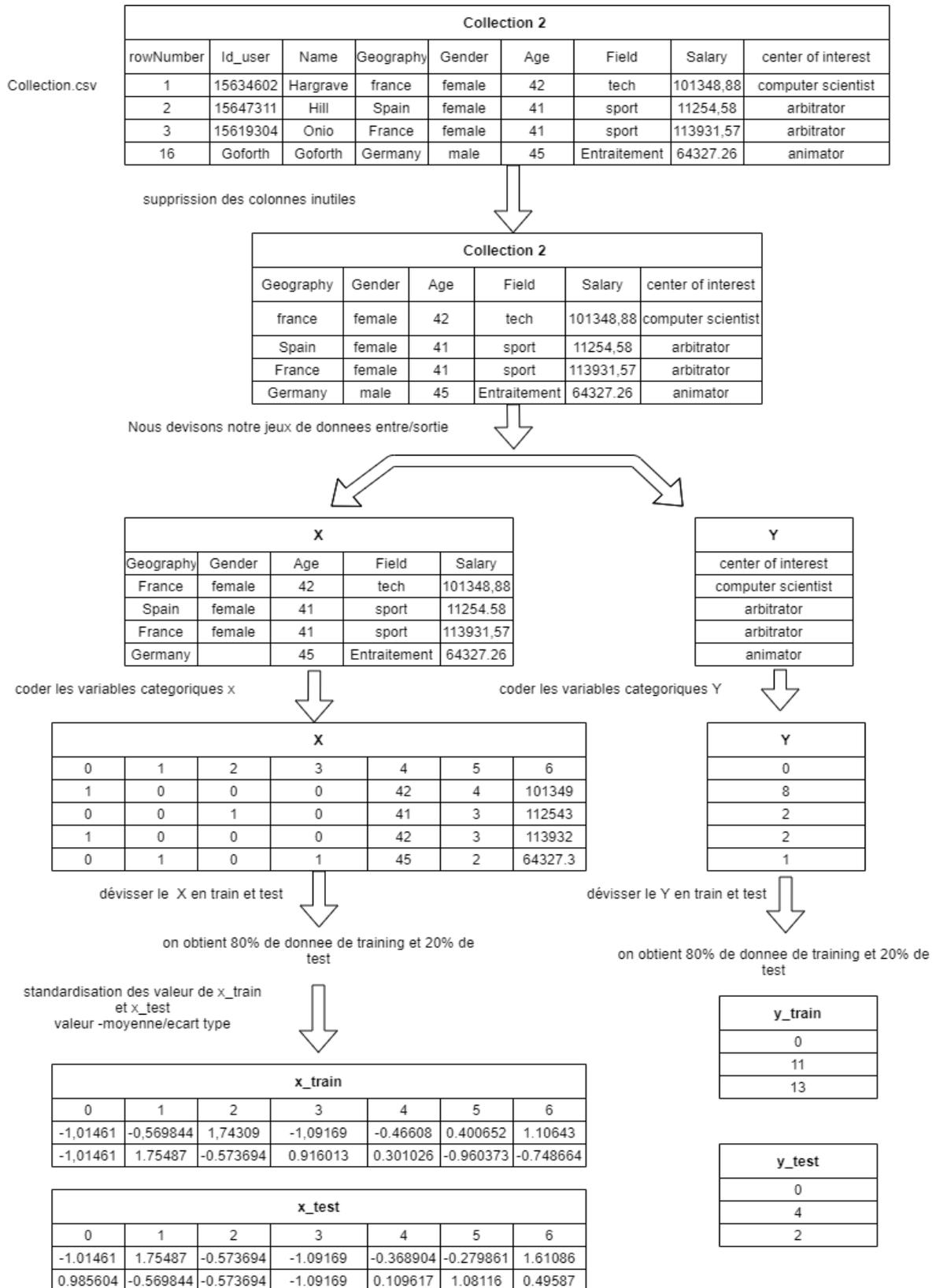


Figure V.4 Prétraitement et représentation de la deuxième collection

V.6 Entraînement et résultats

V.6.1 Première phase de l'approche

Nous avons entraîné nos trois modèles décrits dans la section précédente sur l'ensemble des données d'entraînement de la première collection. Par la suite, nous avons testé les trois modèles sur l'ensemble de données de test. Nous montrons dans ce qui suit les résultats obtenus en effectuant une évaluation en termes de précision et de perte.

V.6.1.1 Résultats obtenus pour le premier modèle (ANN)

Le premier modèle est construit avec la configuration représentée dans la Figure suivante.

Cette figure résume l'architecture du premier modèle que nous avons décrit dans le chapitre précédent.

```
=====  
embedding_1 (Embedding)      (None, None, 64)      320000  
-----  
global_average_pooling1d_1 ( (None, 64)      0  
-----  
dense_2 (Dense)              (None, 130)           8450  
-----  
dense_3 (Dense)              (None, 70)            9170  
-----  
dense_4 (Dense)              (None, 6)              426  
=====  
Total params: 338,046  
Trainable params: 338,046  
Non-trainable params: 0
```

Figure V.5 Prétraitement et représentation de la deuxième collection

Nous présentons ci-dessous les courbes de précision et de perte montrant l'entraînement et de test du premier modèle avec les données de la première collection.

La figure V.6 représente la courbe de précision de l'ensemble d'entraînement en bleu et la précision de l'ensemble de test en orange en fonction du nombre d'epoch.

La figure V.7 représente la courbe de perte de l'ensemble d'entraînement en bleu et la perte de l'ensemble de test en orange en fonction du nombre d'epoch.

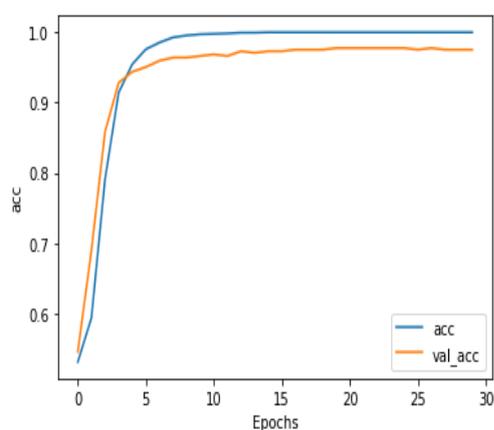


Figure V.6 précision (accuracy) du premier modèle

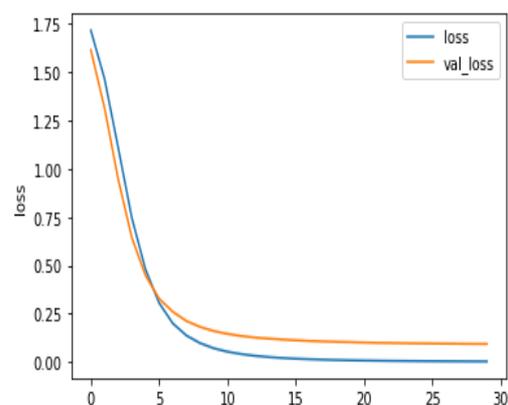


Figure V.7 erreur (loss) du premier modèle

D'après les deux figures, la précision de l'apprentissage et de la validation augmente avec le nombre d'epoch. Ceci reflète qu'à chaque epoch le modèle apprend plus d'informations. De même, l'erreur d'apprentissage et de la validation diminue avec le nombre d'epoch.

Le tableau V.1 montre les résultats obtenus pour le premier modèle.

Précision avec l'ensemble d'entraînement	Précision avec l'ensemble de test
99.49%	97.08%

Tableau V.1 résultats obtenus pour le premier modèle

V.6.1.2 Résultats obtenus pour le deuxième modèle.

Le deuxième modèle est construit avec la configuration représentée dans la figure V.8 .

Cette figure résume l'architecture du deuxième modèle que nous avons décrit dans le chapitre précédent.

```
Layer (type)                Output Shape                Param #
=====
embedding (Embedding)       (None, None, 64)          320000
bidirectional (Bidirectional) (None, 128)                66048
dense (Dense)                (None, 128)                16512
dense_1 (Dense)              (None, 64)                  8256
dense_2 (Dense)              (None, 6)                   390
=====
Total params: 411,206
Trainable params: 411,206
Non-trainable params: 0
```

Figure V.8 Configuration du deuxième modèle

Nous présentons ci-dessous les courbes de précision et de perte montrant l'entraînement et de test du deuxième modèle avec les données de la première collection.

Chapitre 5 : Implémentation et expérimentations

La figure V.9 représente la courbe de précision de l'ensemble d'entraînement en bleu et la précision de l'ensemble de test en orange en fonction du nombre d'epoch.

La figure V.10 représente la courbe de perte de l'ensemble d'entraînement en bleu et la perte de l'ensemble de test en orange en fonction du nombre d'epoch.

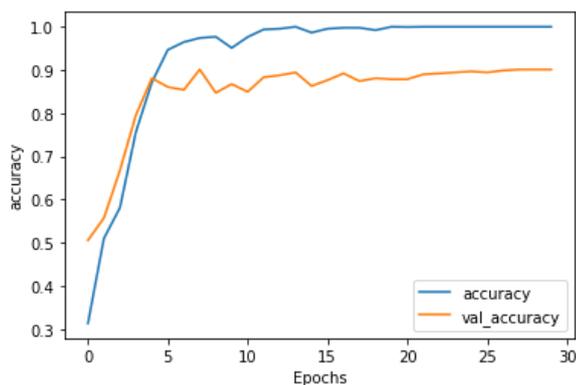


Figure V.9 précision (accuracy) du deuxième modèle

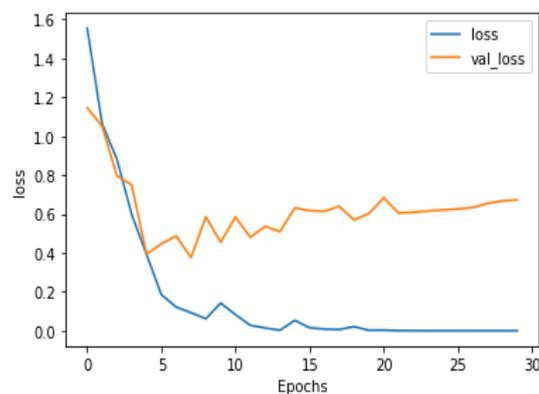


Figure V.10 erreur (loss) du deuxième modèle

D'après les deux figures, la précision de l'apprentissage et de la validation augmente avec le nombre d'epoch.

Ceci reflète qu'à chaque epoch le modèle apprend plus d'informations. De même, l'erreur d'apprentissage et de la validation diminue avec le nombre d'epoch. Mais on remarque qu'il y a un sur-apprentissage, un écart de 10% entre la précision de l'entraînement et la pression de test.

Le tableau V.2 montre les résultats obtenus.

Chapitre 5 : Implémentation et expérimentations

Précision avec l'ensemble d'entraînement	Précision avec l'ensemble de test
99.20%	90,01%

Tableau V.2 résultats obtenus pour le deuxième modèle

V.6.1.3 Une variante du deuxième modèle

Après les résultats obtenus avec le deuxième modèle et dans le but d'améliorer le taux d'apprentissage et d'éviter le sur-apprentissage, nous avons ajouté un régularisateur (Dropout).

Sa configuration est représentée dans la figure suivante.

```
Layer (type)                Output Shape                Param #
=====
embedding_2 (Embedding)     (None, None, 64)          320000
bidirectional_2 (Bidirection (None, 128)          66048
dense_6 (Dense)              (None, 128)                16512
dropout_2 (Dropout)         (None, 128)                0
dense_7 (Dense)              (None, 64)                 8256
dense_8 (Dense)              (None, 6)                  390
=====
Total params: 411,206
Trainable params: 411,206
Non-trainable params: 0
```

Figure V.11 Configuration de la variante du deuxième modèle

Nous présentons ci-dessous les courbes de précision et de perte montrant l'entraînement et de test pour ce modèle avec les données de la première collection.

Chapitre 5 : Implémentation et expérimentations

La figure V.12 représente la courbe de précision de l'ensemble d'entraînement en bleu et la précision de l'ensemble de test en orange en fonction de nombre d'epoch.

La figure V.13 représente la courbe de perte de l'ensemble d'entraînement en bleu et la perte de l'ensemble de test en orange en fonction de nombre d'epoch.

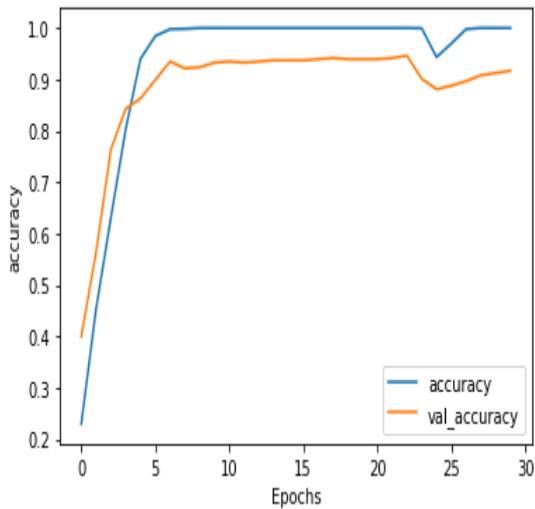


Figure V.12 précision(accuracy) de la variante du deuxième modèle

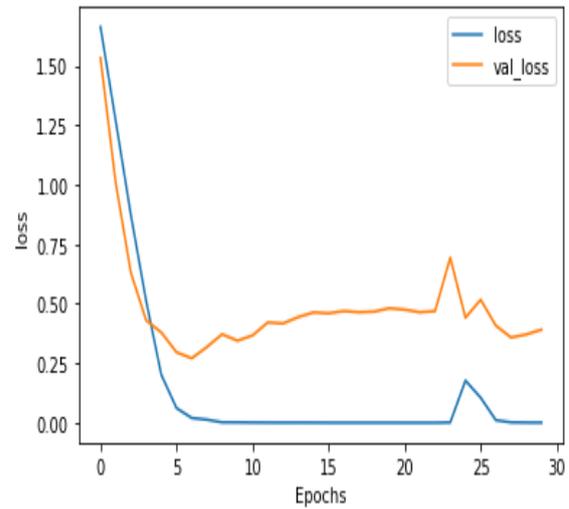


Figure V.13 erreur (loss) de la variante du deuxième modèle

D'après les deux figures, la précision de l'apprentissage et de validation est améliorée par rapport à celle obtenue par le deuxième modèle. On constate que la précision a atteint 94% à l'epoch 23 puis elle a diminué. On déduit donc que le nombre d'epoch influence sur la performance du modèle.

Le tableau V.3 montre les résultats obtenus pour ce modèle.

Précision avec l'ensemble d'entraînement	Précision avec l'ensemble de test
99.50%	92,07%

Tableau V.3 résultats obtenus pour la variante du deuxième modèle

V.6.1.4 Résultats obtenus pour le troisième modèle

Le troisième modèle est construit avec la configuration représentée dans la Figure V.14.

Cette figure résume l'architecture du premier modèle que nous avons décrit dans le chapitre précédent.

```
Layer (type)                Output Shape                Param #
=====
embedding_1 (Embedding)     (None, None, 64)          320000
conv1d_1 (Conv1D)           (None, None, 128)         41088
global_average_pooling1d (Gl (None, 128)                0
dense_2 (Dense)             (None, 64)                 8256
dense_3 (Dense)             (None, 6)                  390
=====
Total params: 369,734
Trainable params: 369,734
Non-trainable params: 0
```

Figure V.14 Configuration de la variante du troisième modèle.

Nous présentons ci-dessous les courbes de précision et de perte montrant l'entraînement et de test du troisième modèle avec les données de la première collection.

Chapitre 5 : Implémentation et expérimentations

La figure V.15 représente la courbe de précision de l'ensemble d'entraînement en bleu et la précision de l'ensemble de test en orange en fonction de nombre d'epoch.

La figure V.16 représente la courbe de perte de l'ensemble d'entraînement en bleu et la perte de l'ensemble de test en orange en fonction de nombre d'epoch.

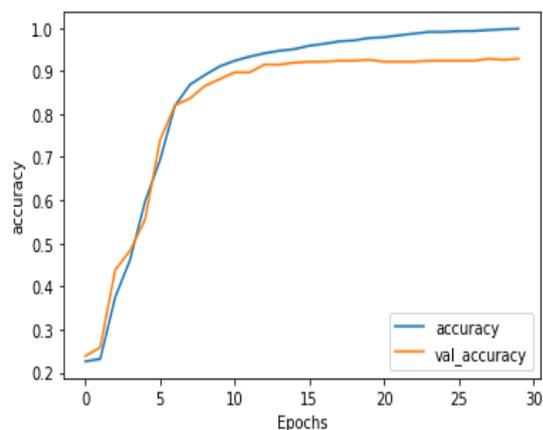


Figure V.15 précision (accuracy) du troisième modèle

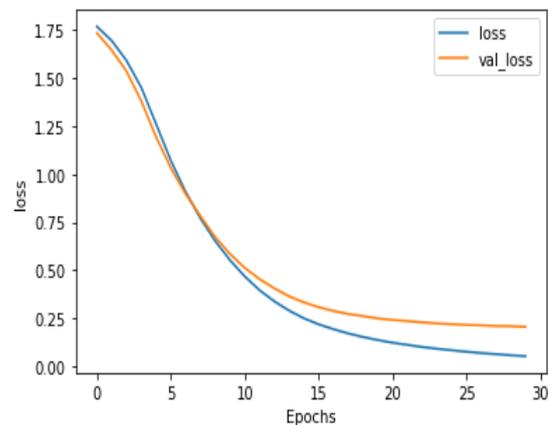


Figure V.16 erreur(loss) du troisième modèle

D'après les deux figures, la précision de l'apprentissage et de la validation augmente avec le nombre d'epoch. Ceci reflète qu'à chaque epoch le modèle apprend plus d'informations. De même, l'erreur d'apprentissage et de la validation diminue avec le nombre d'epoch.

Le tableau V.4 montre les résultats obtenus pour le troisième modèle.

Précision avec l'ensemble d'entraînement	Précision avec l'ensemble de test
99.80%	97.50%

Tableau V.4 résultats obtenus pour la variante du troisième modèle.

V.6.1.5 Discussion et comparaison des résultats

Les résultats obtenus pour les trois modèles sont comparés dans le tableau V.5 ci-dessous.

le modèle	nombre d'epoch	Précision de l'entraînement	Précision de test
modèle 1	30	99.49%	97,08%
modèle 2	30	99,20%	90,01%
variante modèle 2	30	99,50%	92.07%
modèle 3	30	99,80%	97,50%

Tableau V.5 Comparaison des résultats obtenus pour les trois modèles de la première collection

Nous observons que les résultats obtenus sont assez similaires pour les trois modèles (1, 2 et 3). On constate que différents paramètres influence sur les performances du modèle, à savoir l'architecture des réseaux, le nombre d'epoch, la régularisation qui permet d'éviter le sur apprentissage.

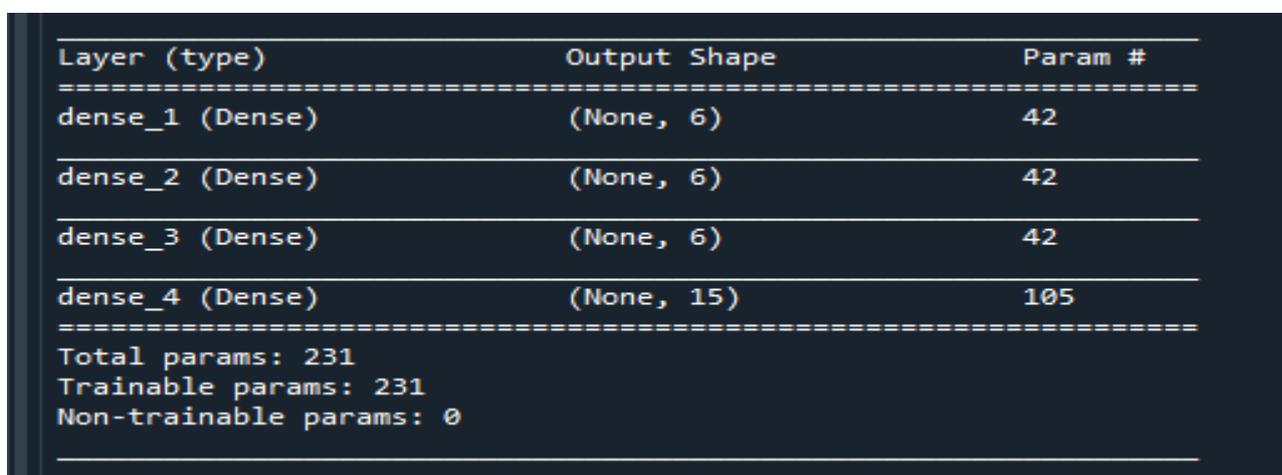
D'après ces résultats, on constate que le troisième modèle est le plus performant.

V.6.2 Deuxième phase de l'approche

V.6.2.1 Résultats obtenus pour le premier modèle

Le premier modèle est construit avec la configuration représentée dans la Figure suivante.

Cette figure résume l'architecture du premier modèle que nous avons décrit dans le chapitre précédent.



```
Layer (type)                Output Shape                Param #
-----
dense_1 (Dense)             (None, 6)                   42
dense_2 (Dense)             (None, 6)                   42
dense_3 (Dense)             (None, 6)                   42
dense_4 (Dense)             (None, 15)                  105
-----
Total params: 231
Trainable params: 231
Non-trainable params: 0
```

Figure V.17 Configuration du premier modèle

Nous présentons ci-dessous les courbes de précision et de perte montrant l'entraînement du premier modèle avec le jeu de données de la deuxième collection.

La figure V.18 représente la courbe de précision de l'ensemble d'entraînement en bleu et la précision de l'ensemble de test en orange en fonction de nombre d'epoch.

La figure V.19 représente la courbe de perte de l'ensemble d'entraînement en bleu et la perte de l'ensemble de test en orange en fonction de nombre d'epoch.

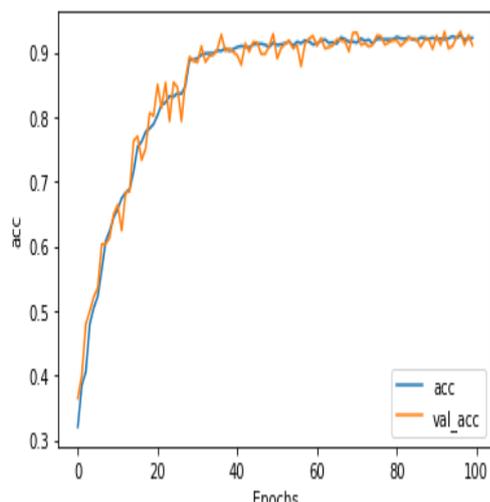


Figure V.18 précision (accuracy) du premier modèle

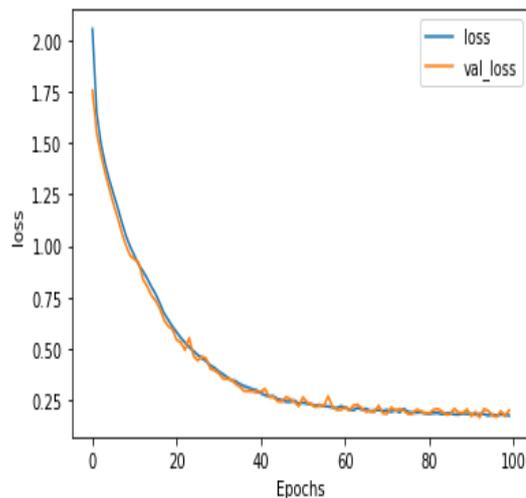


Figure V.19 erreur(loss) du premier modèle

D'après les deux figures, la précision de l'apprentissage et de la validation augmente avec le nombre d'époch. Ceci reflète qu'à chaque epoch le modèle apprend plus d'informations. De même, l'erreur d'apprentissage et de la validation diminue avec le nombre d'époch.

Mais les performances d'apprentissage ne sont pas satisfaisantes.

Le tableau V.6 montre les résultats obtenus pour le premier modèle.

Précision avec l'ensemble d'entraînement	Précision avec l'ensemble de test
79,85%	78,75%

Tableau V.6 résultats obtenus pour le premier modèle.

V.6.2.2 Résultats obtenus pour le deuxième modèle

Le deuxième modèle est construit avec la configuration représentée dans la Figure suivante

Cette figure résume l'architecture du premier modèle que nous avons décrit dans le chapitre précédent.

```
Layer (type)                Output Shape                Param #
=====
dense_12 (Dense)            (None, 50)                  300
dense_13 (Dense)            (None, 40)                  2040
dense_14 (Dense)            (None, 20)                  820
dense_15 (Dense)            (None, 15)                  315
=====
Total params: 3,475
Trainable params: 3,475
Non-trainable params: 0
```

Figure V.20 Configuration du deuxième modèle

Nous présentons ci-dessous les courbes de précision et de perte montrant l'entraînement et de test du deuxième modèle avec le jeu de données de la deuxième collection.

La figure V.21 représente la courbe de précision de l'ensemble d'entraînement en bleu et la précision de l'ensemble de test en orange en fonction de nombre d'epoch.

La figure V.22 représente la courbe de perte de l'ensemble d'entraînement en bleu et la perte de l'ensemble de test en orange en fonction de nombre d'epoch.

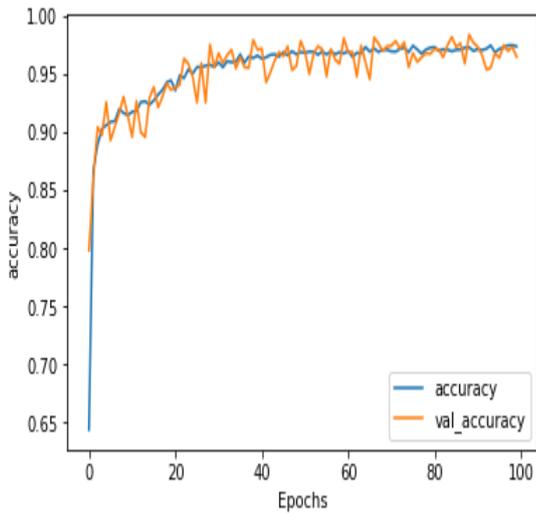


Figure V.21 précision (accuracy) du deuxième modèle

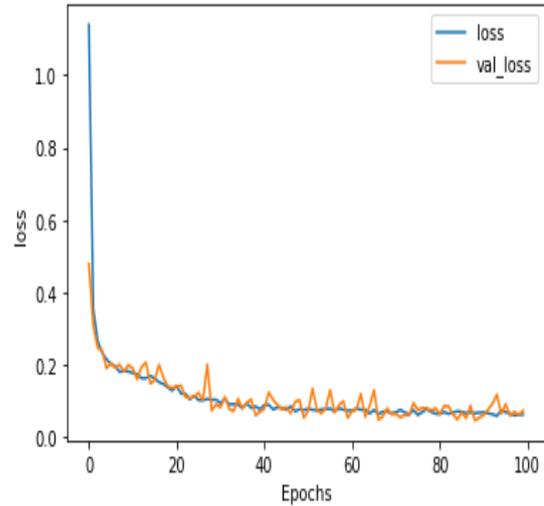


Figure V.22 erreur (loss) du deuxième modèle

D'après les deux figures, la précision de l'apprentissage et de la validation augmente avec le nombre d'epoch. Ceci reflète qu'à chaque epoch le modèle apprend plus d'informations. De même, l'erreur d'apprentissage et de la validation diminue avec le nombre d'epoch.

D'après les résultats obtenus, nous pouvons déduire que plus le réseau est grand plus l'algorithme apprend et prédit mieux.

Le tableau V.7 montre les résultats obtenus pour le troisième modèle.

Précision avec l'ensemble d'entraînement	Précision avec l'ensemble de test
97,39%	96,50%

Tableau V.7 résultats obtenus pour le deuxième modèle.

V.6.2.3 Discussion et comparaison des résultats

Les résultats obtenus pour les deux modèles sont comparés dans le tableau V.8 ci-dessous

le modèle	nombre d'epoch	Précision de l'entraînement	Précision de test
modèle 1	100	79,85%	78,75%
modèle 2	100	97,39%	96,50%

Tableau V.8 Comparaison des résultats obtenue des deux modèles avec la deuxième collection.

D'après ces résultats, Nous pouvons constater que l'utilisation d'une architecture plus grande permet au modèle d'apprendre mieux et de prédire mieux.

V.7 Conclusion

Dans ce chapitre, nous avons présenté en premier lieu, les différents outils et environnement de développement utilisés pour l'implémentation des modèles de notre approche. Ensuite, nous avons décrit les collections d'entraînement et de test utilisées. Puis, nous avons présenté la partie implémentation de notre approche, où nous avons expliqué en détail les étapes de prétraitement et de représentation. Enfin, nous avons discuté les résultats obtenus après l'entraînement de chaque modèle sur nos collections.

L'évaluation de notre approche, réalisée sur les modèles proposés ci-dessus sur les deux collections ont mené aux constatations qui suivent : le choix de l'architecture, le nombre de couches, le nombre d'unités et le nombre d'itérations d'apprentissage influencent sur les résultats de l'entraînement et de test.

Chapitre 5 : Implémentation et expérimentations

En effet, les expérimentations ont montré que le troisième modèle de la première collection et le deuxième modèle de la deuxième collection sont idéals pour avoir un meilleur résultat d'entraînement et de test.

Conclusion générale

La croissance considérable de volume des données ainsi que la diversité des besoins des utilisateurs rendaient l'accès à une information pertinente de plus en plus difficile. D'où le besoin de personnaliser l'information recherchée par l'utilisateur, en fonction de son profil.

Dans ce travail, nous nous sommes intéressés à la construction du profil utilisateur. Nous avons défini ce profil par son centre d'intérêt que nous inférons automatiquement à l'aide des réseaux de neurones.

Pour inférer automatiquement les centres d'intérêts, nous avons proposé différents modèles de réseaux de neurones que nous avons évalué en utilisant des collections de données. En premier lieu nous avons inféré un centre d'intérêt à partir des documents issus de la collection bbc.

Une fois le centre d'intérêt est inféré nous l'avons enrichi avec d'autres centres d'intérêts plus spécifiques, en l'utilisant comme entrée d'un réseau de neurones avec d'autres données liées à l'utilisateur issue de la collection que nous avons créé.

Les différentes expérimentations réalisées ont montré l'intérêt des modèles adoptés, avec un résultat de test très élevé dans le troisième modèle de la première collection et le deuxième modèle de la deuxième collection.

Comme perspectives à notre travail nous envisageons de :

- Tester nos modèles sur d'autres collections de données.
- Développer des modèles d'évolution du profil utilisateur.
- Intégration du profil défini dans un système de recherche d'information personnalisée.

Bibliographie

[Amato and Straccia, 1996]

Amato G. and Straccia U. User profile modeling and applications to digital libraries. In European Conference on Digital Libraries (ECDL), Paris. Springer. 37, 38,1999.

[Ammar, 2007]

Ammar M.Y., Mise en oeuvre de réseaux de neurones pour la modélisation de cinétique réactionnelles en vue de la transposition batch/continu. Tunis : Université de Tunis, 2007. Thèse de Doctorat.

[Amraoui, 2011]

Amraoui A., Amélioration de la fiabilité du lien sans fil pour un terminal radio cognitive mobile. Tlemcen : Université de Tlemcen, 2011. Thèse de Doctorat.

[Asradj, 2009]

Asradj Z., identification des systèmes non-linéaires par les réseaux de neurones. bédjaia : Université Abd Erahmane Mira, 2015. Mémoire de fin d'étude .

[Azizi, 2018]

Azizi A., Classification automatique de textes arabes supervisée par l'ontologie lexicale wordnet. Msila : Université Mohamed Boudiaf Msila, 2017 /2018. Mémoire de fin d'étude.

[Baeza-Yates and Ribeiro-Neto, 1999]

Baeza-Yates R. and Ribeiro-Neto B. , "Modern information retrieval". ACM Addison-Wesley. 1999.

[Barigou, 2013]

Barigou F., Contribution à la Catégorisation de Textes et à l'Extraction d'Information. Oran : Université d'Oran, 2012/2013. Thèse doctorat.

[Belkin and Croft, 1992]

Belkin N.J., Croft W.B., Information filtering and information retrieval: two sides of the same coin, Communications of the ACM, vol. 35, n° 12, p. 29-38, décembre 1992.

[Bendaoud, 2014]

Bendaoud Y., Prédiction des résistances mécaniques des bétons à base des ciments composés en utilisant les réseaux neurones artificiel. Constantine : Université Constantine1, 2014. Thèse de Magister.

[Bengio et al., 2014]

Bengio Y., Chung J., Gulcehre C. and Cho K.H. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling, 2014.

[Brajnik et al., 1996]

Brajnik G., Mizzaro S., and Tasso C., Evaluating user interfaces to information retrieval systems : a case study on user support. In Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 128.136, Zurich, 1996.

[Chaoui, 2002]

Chaoui H., Conception et comparaison de lois de commande adaptative à base des réseaux de neurones pour une articulation flexible avec non-linéarité dure. québec : l'universités du québec à trois-rivières, 2002. Mémoire de fin d'étude.

[Chen and Sycara, 1998]

Chen L. and Sycara K., Webmate : A personal agent for browsing and searching. In Proceedings of the 2nd international conference on autonomous agents and multi agent systems, Minneapolis, pages 10–13, 1998.

[Cosijn and Ingwersen, 2000]

Cosijn E. and Ingwersen P., "Dimensions of relevance". *Information Processing & Management (IP&M)*, 36(4), , pp:533–550,2000.

[Dendani, 2010]

Dendani H., Modélisation et détection des modes de défaillance dans un système complexe par les méthodes graphiques:Application au réseau électrique Sonelgaz. Annaba : Université Badji Mokhtar, 2010. Mémoire de fin d'etude .

[Fusiello et al., 2009]

Fusiello A., Toldo R. and Castellani U., A bag of words approach for 3d oobject categorization. france : Springer, 2009. Comptes rendus de conférence .

[Gammerman et al., 1998]

Gammerman A., Vovk V., and Vapnik V., Learning by transduction In Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence. Californie : Morgan Kaufmann Publishers Inc, 1998. Comptes rendus de conférence .

[Gauch et al., 2003]

Gauch S., Chafiee J., and Pretschner A., Ontology-based personalized search and browsing. *Web Intelli. and Agent Sys*, 1(3-4) :219#234, 2003.

[Gauch et al., 2007]

Gauch S., Speretta M., Chandramouli A., Micarelli A., « User Profiles for Personalized Information Access », dans Brusilovsky P., Kobsa A., Nejdl W. (dirs.), *The Adaptive Web*, Springer Berlin Heidelberg (Lecture Notes in Computer Science), p. 54 89, 2007.

[Gentili et al., 2003]

Gentili G., Micarelli A., and Sciarrone F., Infoweb: An adaptive information filtering system for the cultural heritage domain. *Applied Artificial Intelligence*, 17(8-9) :715#744, 2003.

[Hammoud, 2014]

Hammoud D., Apprentissage Automatique dans un Agent. constantine : université Mentouri Constantine, 2014. Thèse de Doctorat.

[Kostadinov, 2003]

Kostadinov D., La personnalisation de l'information, définition de modèle de profil utilisateur. Rapport de dea. Master's thesis, Université de Versailles, France, 2003.

[Kostadinov, 2007]

Kostadinov D., Personnalisation de l'information : une approche de gestion de profils et de reformulation de requêtes. PhD thesis, l'universite de versailles saint-quentin-en-yvelines, Décembre 2007.

[Koutrika and Ioannidis, 2004]

Koutrika G., Ioannidis Y., Personalization of Queries in Database Systems, In: Proceedings of the 20th International Conference on Data Engineering, Boston, Massachusetts, USA, April, 2004.

[Koutrika and Ioannidis, 2005]

Koutrika, G. and Ioannidis Y., A unified user-profile framework for query disambiguation and personalization. Workshop on New Technologies for Personalized Information Access in conjunction with the 10th International User Modeling, pages pp 44#53, 2005.

[Krief and Benmammar, 2006]

Krief F. and Benmammar B., Resource Management for End-to-End QoS in a Mobile Environment. Montréal : International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob 2006), 2006. Comptes rendus de conférence.

[Larochelle, 2008]

Larochelle H., Etude de techniques d'apprentissage non-supervisé pour l'amélioration de l'entraînement supervisé de modèles connexionnistes. Montréal : Université de Montréal, 2008. Thèse Doctorat

[Laur, 2017]

Laur M., Anticipation des changements des notes des obligations des portefeuilles d'un assureur par méthode de machine learning. Paris : Université Dauphine, 2017. Mémoire de fin d'étude.

[Lelu and Francois, 1992]

Lelu A. and Francois, " Information retrieval based on neural unsupervised extraction of thematic fuzzy clusters". In Proc. of the International Conference, Neural Networks and their Applications, pp: 93–104, 1992.

[Lewis, 1998]

Lewis D., Naive (Bayes) at forty: The independence assumption in information retrieval. In European conference on machine learning. Berlin : Springer, 1998. Comptes rendus de conférence .

[Lie Ba and Kingma, 2015]

Lie Ba J. and Kingma D.P. 3rd International Conference for Learning Representations. San Diego : edition San Diego, 2015. Comptes rendus de conférence.

[Lieberman, 1995]

Lieberman H., Letizia : An agent that assists web browsing. In Proceedings of the International Joint Conference on Artificial Intelligence, Montreal, August 1995.

[Marzouka, 2009]

Merzouka N., Etude des performances des réseaux de neurones dynamiques à représenter des systèmes réels : une approche dans l'espace d'état. Setif : Université de Setif1, 2009. Mémoire de fin d'étude .

[Mctear, 1993]

Mctear M.F., User modelling for adaptive computer systems: a survey of recent developments." Artificial Intelligence Review 7: 157-184, 1993.

[Mokri, 2017]

Mokri M.Z., classification des images avec les réseaux de neurones convolutionnels. Tlemcen : Université Abou Bakr Belkaid, 2017. Mémoire de fin d'étude .

[Moualek, 2017]

Moualek D.Y., Deep Learning pour la classification des images. Tlemcen : Université Abou Bakr Belkaid– Tlemcen, 2016/2017. mémoire de fin d'etude .

[Nene, 2017]

Nene S., Deep Learning for Natural Language Processing. International Research Journal of Engineering and Technology (IRJET).2017. Dept. of Computer Engineering, VESIT, Maharashtra,India.

[Patterson and Gibson, 2017]

Patterson J. and Gibson A. ,2017. Deep Learning A Practitioner's Approach,1 ère (Ed), O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472 , Mike Loukides & Tim McGovern, 532 p, (pp. 28)

[Rimouche and Hachemi, 2015]

Rimouche N.E. and Hachemi H., Amélioration du produit scalaire via les mesures de similarités sémantiques dans le cadre de la catégorisation des textes. Tlemcen : Université Abou Bakr Belkaid Tlemcen, 2015. Mémoire de fin d'études.

[RISCH, 2017]

RISCH J., Enrichissement des Modèles de Classification de Textes Représentés par des Concepts. Reims : Université De Reims Champagne-Ardenne, 2017. Thèse Doctorat.

[Robertson and Sparck Jones, 1976]

Robertson S. and Sparck Jones K., Relevance weighting for search terms. Journal of The American Society for Information Science, 27(3) :129–146, 1976.

[Robertson et al., 1995]

Robertson S., Walker S., Sparck Jones M. and al. Okapi at trec-3. In Second Text Retrieval Conf (TREC-3), pages 109.26, 1995.

[Robertson et al., 1999]

Robertson S., Walker S., and Beaulieu M., Okapi at trec-7 : Automatic ad hoc, filtering, vlc and interactive track, 1999.

[Robertson et al., 2000]

Robertson S., Walker S., and Beaulieu M., Experimentation as a way of life : Okapi at trec. Information Processing & Management, 36(1) :95.108, 2000.

[Rohini and Ambati, 2005]

Rohini U. and Ambati V., A collaborative filtering based re-ranking strategy for search in digital libraries. In ICADL, pages 194#203, 2005.

[Rosenfeld et al., 1988]

Rosenfeld E., Anderson J. and Pellionisz A., Neurocomputing : Foundations of research, MIT Press. Cambridge : Second printing, 1988. Comptes rendus de conférence

[Salton et al., 1973]

Salton, Gerard, Yang and C. S. On the specification of term values in automatic indexing. Journal of Documentation, 29(4) :351#372, 1973.

[Salton et al., 1986]

Salton, Gerard and Michael J. McGill , "Introduction to Modern Information Retrieval".New York, NY, USA: McGraw-Hill, Inc. ISBN 0070544840.1986.

[Salton, 1971]

Salton G., The SMART Retrieval System—Experiments in Automatic Document Processing. Prentice-Hall Inc, NJ, 1971.

[Schmidhuber and Hochreiter, 1997]

Schmidhuber J. and Hochreiter S., Neural Computation. Allemagne : Université technique de Munich, 1997.

[Shen et al., 2005]

Shen X., Tan B., and Zhai C., Implicit user modeling for personalized search. In CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management, pages 824#831, New York, NY, USA, 2005. ACM Press.

[Singhal, 1997]

Singhal A., Term weighting revisited. PhD thesis, PHD of Cornell University, 1997.

[Stefani and Strappavara, 1998]

Stefani A. and Strappavara C., Personalizing access to web sites: The siteif project. In Proceedings of the 2nd Workshop on Adaptive Hypertext and Hypermedia, Pittsburgh, June 20 24 1998.

[Tamine and Calabretto, 2008]

Tamine L. and Calabretto S., Recherche d'information sur le web : état des lieux et perspectives, chapter Recherche d'information contextuelle et web. Number 7. Editions Hermes, sous la direction de Mohand Boughanem et Jacques Savoy, 2008.

[Tellie, 2010] Tellie I., L'apprentissage automatique : Preface. Orléan : Université d'Orléan, 2010.

[Trenkle and Cavnar, 1994]

Trenkle J.M. and Cavnar W.B., N-gram-based text categorization. Michigan : Ann Arbor MI, 1994. Comptes rendus de conférence.

[Van Rijsbergen, 1986]

Van Rijsbergen C., A non-classical logic for information retrieval. The computer Journal, 29(6) :481–485, 1986.

[Vedantam et al., 2016]

Vedantam R., Kottur S. Parikh D., Learning visually grounded word embeddings using abstract. la chine : Dans les actes de CVPR, 2016. Comptes rendus de conférence .

[Widyantoro et al., 1999]

Widyantoro D., Yin J., El Nasr M., Yang L., Zacchi A., and Yen J., Alipes : A swift messenger in cyberspace. In Proceedings of Spring Symposium Workshop on Intelligent Agents in Cyberspace, pages 62–67, Stanford, March 22-24 1999.

[Wu et al., 2008]

Wu D., Zhao D., and Zhang X. An adaptive user profile based on memory model. In Proceedings of the 2008 The Ninth International Conference on Web-Age Information Management, WAIM '08, pages 461#468, Washington, DC, USA. IEEE Computer Society, 2008.

[Zemirli , 2008]

Zemirli W. N. Modèle d'accès personnalisé à l'information basé sur les Diagrammes d'influence intégrant un profil utilisateur évolutif. PhD thesis, Université Paul Sabatier de Toulouse III, 2008.

[Zhou and Rossetto, 2019]

Zhou W. and Rossetto A.M., Improving Classification with CNNs using Wavelet Pooling with Nesterov-Accelerated Adam, P84-93. Lowell : Université Massachusetts Lowell, 2019. Comptes rendus de conférence.