République Algérienne Démocratique et Populaire
Ministère de l'enseignement Supérieur et de la Recherche Scientifique
Université de Mohamed El Bachir El Ibrahimi de Bordj Bou Arréridj
Faculté des Mathématiques et d'Informatique
Département d'informatique

MEMOIRE

Présenté en vue de l'obtention du diplôme

Master en informatique

Spécialité : Technologie de l'information et de communication (TIC)

THEME

Catégorisation multi-étiquette des textes arabes

Présenté par **:**

Bendib Yassamine

Bouaziz Meriem

Soutenu publiquement le : 14/06/2025
Devant le jury composé de :

Président : Dr. Boutouhami Sara

Examinateur : Dr. Benmessahel Ilyes

Encadreur : Dr.Nouioua Mourad

2024/2025

اهداء

{وَآخرُ دَعواهُمْ أَنِ الْحَمدُ لِلَّهِ ربِّ الْعالَمِينَ}
أهدي هذا الإنجاز الذي انتظرته طويلًا، إلى من كان لهم الأثر الأعمق في رحلتي:

إلى الذي زين اسمي بأجمل الألقاب من دعمني بلا حدود وأعطاني بلا مقابل إلى من علمني أن الدنيا كفاح وسلاحها العلم والمعرفة الى من غرس في روحي مكارم الأخلاق داعمي الأول في مسيرتي ملاذي بعد الله..
إلى فخري و اعتزازي

"والدي"

إلى من جعل الله الجنة تحت أقدامها واحتضنني قلبها قبل يدها وسهّلت لي الشدائد بدعائها إلى القلب الحنون والشمعة التي كانت لي في الليالي المظلمات
سر قوتي ونجاحي ومصباح دربي الى وهج حياتي

"والدتي"

أهدي تخرجي إلى ملهمي نجاحي: من ساندني بكل حب عند ضعفي وأزاح عن طريقي المتاعب، سندي وكتفي الذي أستند عليه دائمًا طالما كانوا الظل لهذا النجاح.

"أخوتي"

من تقاسمتم معي اللحظات الحلوة والمرة، وكنتم لي إخوة قبل أن تكونوا أصدقاء.

" أصدقائي"

و لا أنسى رفيق روحي سندي وشريك أحلامي، الذي شاركني خطوات هذا الطريق وهون التعب

" خطيبي ".

و لمن علمني حرفًا يومًا ما و أخذت من علمهم و حضرت دروسهم و شاركوني خبراتهم.

"اساتذتي الكرام"

ياسمين بن الذيب

اهداء

إلى من أوصانا بهم الرحمن حين قال

{ وَاخْفِضْ لَهُمَا جَنَاحَ الذُّلِّ مِنَ الرَّحْمَةِ وَقُل رَّبِّ ارْحَمْهُمَا كَمَا رَبَّيَانِي صَغِيرًا}

إليكما يا من كان دعاؤكما سلاحي وصبركما زادي وحبكما سندي

"أمي وأبي الغاليين"

إلى من كانوا السند والدعم والكتف الذي أتكئ عليه عندما تميل بي الحياة

"إخوتي الأحبة"

إلى أصدقاء المواقف لا السنين، إلى رفاق خطوات النجاح بدءًا من أول خطوة

وانتهاءً بآخر خطوة

"أصدقاؤنا الأوفياء"

مريم بوعزيز

Remerciement

Nous tenons tout d'abord à exprimer notre gratitude envers ALLAH, qui nous a accordé la santé, le courage, la persévérance et la sagesse nécessaires pour mener à bien ce travail.

Nous souhaitons aussi exprimer nos sincères remerciements à notre superviseur, Dr.**Nouioua Mourad**, pour son soutien constant, ses conseils éclairés et son accompagnement tout au long de réalisation de ce mémoire.

Nous adressons nos remerciements aux membres du jury, pour avoir accepté d'évaluer ce mémoire.

Nous souhaitons ainsi exprimer notre reconnaissance envers nos proches, nos familles et nos amis, pour leur soutien inconditionnel, leurs encouragements constants et leur compréhension tout au long de ce parcours académique.

Enfin, nous tenons à remercier tous ceux qui, de près ou de loin, ont contribué à la réalisation de ce mémoire. Leur apport, qu'il soit intellectuel, moral ou logistique, a joué un rôle important dans la concrétisation de ce travail.

Nous sommes profondément reconnaissantes envers toutes ces personnes qui ont joué un rôle essentiel dans la réussite de ce mémoire Leur soutien indéfectible et leurs conseils précieux resteront gravés dans notre mémoire.

Merci du fond du cœur.

Résumé

Ce mémoire porte sur la classification multi-étiquettes des textes arabes à l'aide de techniques d'apprentissage profond. La langue arabe présente plusieurs défis, notamment une complexité morphologique élevée, une richesse lexicale importante et une grande variabilité dialectale. Ce travail se concentre sur l'utilisation de réseaux de neurones profonds, en particulier le modèle convolutionnel (CNN), BiLSTM et le modèle pré-entraîné AraBERT, afin d'améliorer la précision du classement.

Le mémoire aborde les étapes clés du traitement automatique de la langue arabe, les méthodes de représentation des textes, ainsi que les métriques d'évaluation adaptées à la classification multi-étiquettes. Une attention particulière est portée à l'apprentissage par transfert, qui permet de tirer parti de la puissance des modèles linguistiques pré-entraînés pour la langue arabe.

Mots-clés : Classification multi-étiquettes, Textes arabes, Apprentissage profond, Apprentissage par transfert

Abstract

This thesis focuses on the multi-label classification of Arabic texts using deep learning techniques. The Arabic language presents several challenges, including high morphological complexity, rich vocabulary, and significant dialectal variation. This work focuses on the use of deep neural networks, particularly the Convolutional Neural Network (CNN), BiLSTM and the pre-trained model AraBERT, to improve classification accuracy.

The thesis discusses the key stages of Arabic natural language processing, text representation methods, and evaluation metrics appropriate for multi-label classification. Special attention is given to transfer learning, which leverages the power of pre-trained language models for the Arabic language.

Keywords: Multi-label classification, Arabic texts, Deep learning, Transfer learning

ملخص

يركّز هذا البحث على تصنيف النصوص العربية متعددة التسميات باستخدام تقنيات التعلم العميق. تواجه اللغة العربية عدّة تحديات، من بينها التعقيد الصرفي العالي، وغِنى المعجم، والتنوّع الكبير في اللهجات. يركّز هذا العمل على استخدام الشبكات العصبية العميقة، وخصوصًا نموذج الشبكة العصبية الالتفافية (CNN) وشبكة الذاكرة طويلة وقصيرة المدى ثنائية الاتجاه(BiLSTM) ، والنموذج المدرب مسبقًاAraBERT ، من أجل تحسين دقة التصنيف.

يتناول هذا البحث المراحل الأساسية للمعالجة الآلية للغة العربية، وأساليب تمثيل النصوص، بالإضافة إلى مقاييس التقييم المناسبة لتصنيف النصوص متعددة التسميات. كما يتم التركيز بشكل خاص على التعلم بالنقل، الذي يتيح الاستفادة من قوة النماذج اللغوية المدربة مسبقًا للغة العربية.

الكلمات المفتاحية: تصنيف متعدد التسميات، نصوص عربية، التعلم العميق، التعلم بالنقل

Table des matières

Introduction Générale 1

1. Contexte Générale : 1
2. Problématique et Objectifs 1
3. Organisation du Mémoire 2

Chapitre 1 : La catégorisation des documents arabe 3

1. Introduction 3
2. Définition de la catégorisation des textes 3
3. Difficultés et défis de la catégorisation des textes écrits en langue arabe 3
   - 3.1 Complexité morphologique 3
   - 3.2. Absence de Voyelles Courtes (Diacritiques) 4
   - 3.3 Manque de corpus annotés et de ressources 4
   - 3.4 Racines et dérivations 4
4. Approches de catégorisation de documents pour la langue arabe 5
   - 4.1 Approches classique 5
   - 4.2 Approches basées sur le traitement automatique de langues naturelles 5
     - 4.2.1 Approches basées sur l'apprentissage automatique (Machine Learning) 6
     - 4.2.2 Approches basées sur l'apprentissage profond (Deep Learning) 7
5. Processus de catégorisation des textes 8
   - 5.1 Prétraitement des textes 8
   - 5.2 Représentation des Textes 9
   - 5.3 Classification des textes 10
     - 5.3.1 Choix des Classificateurs 10
     - 5.3.2 Evaluation des performances des classifieurs 10
   - 5.4 Application de classifier choisis pour la catégorisation des nouveaux textes 10
6. Représentation des textes 10
   - 6.1 TF-IDF 10
   - 6.2 Word2Vec 11
   - 6.3 FastText 11
7. Classification 11
   - 7.1 Classification Multi-classe 11
   - 7.2 Classification Multi- étiquette 11
8. Les approches de transformation du problème de classification multi-étiquette 12

8.1 Pertinence binaire (Binary Relevance BR) 12
8.2 Ensemble de puissances d'étiquettes (Label Powerset LP) 12
8.3 Chaîne de classificateurs (Classifier chains CC) 13
9. Métriques d'évaluation pour la classification multi-étiquette 13
9.1 Coût de Hamming (Hamming Loss) 13
9.2 Accuracy 14
9.3 Précision 14
9.4 Rappel 14
9.5 F-score 14
Conclusion 15
Chapitre 2 : L'apprentissage profond pour la classification des textes 16
1. Introduction 16
2. Réseaux de Neurones Artificiels (RNA) 16
3. Les principaux composants du réseau de neurones 16
3.1 Neurones 16
3.2 Couches (groupement de neurones) 16
3.3 Poids et biais 17
3.4 Fonction d'activation 17
3.4.1 Types de fonctions d'activation 18
4. Les méthodes d'apprentissage profond (deep learning) utilisée 18
4.1 Les réseaux de neurones récurrents (RNN) 19
4.2 Long-Short Term Memory (LSTM) 19
4.2.1 Réseaux Bi-directionnels LSTM (BiLSTM) 20
4.3 Réseaux de neurones convolutifs (CNN) 20
5. Transfer learning et modèle pré-entrainé 20
5.1 Transfer learning 20
5.1.1 Fonctionnement de l'apprentissage par transfert 21
5.1.2 Les principaux avantages du transfer learning 22
5.1.3 Utilisation du transfert learning dans le traitement du langage naturel 22
5.2 Modèle pré-entraîné 23
5.2.1 Avantages de l'utilisation de modèles pré-entraînés 23
5.2.2 Applications courantes des modèles pré-entraînés 23
5.2.3 Le modèle pré-entraîné AraBERT (Arabic Bidirectional Encoder Representations from Transformers) 24

Conclusion 24

Chapitre 3 : Modélisation et implémentation 25

1. Introduction 25
2. Architecture du système 25
3. Fonctionnalités du système de catégorisation des documents 26
   - 3.1 Description de base de donné 26
     - 3.1.1 Visualisation des bigrammes fréquents 27
   - 3.2 Nettoyage et prétraitement de données 28
   - 3.3 Catégorisation des documents 29
     - 3.3.1 Répartition des données 29
     - 3.3.2 Classification des documents 29
   - 3.4 L'évaluation 29
4. Environnement de travail et outils d'implémentation 29
   - 4.1 Matériel 29
   - 4.2 Environnement de programmation 30
     - 4.2.1 Langage de programmation 30
     - 4.2.2 Éditeur de code 30
     - 4.2.3 Librairies et bibliothèques Python intégrés 31
   - 5. Analyse exploratoire de données 33
     - 5.1 Diagramme de Venn pour le chevauchement des catégories 33
     - 5.2 Matrice de cooccurrence des étiquettes 34
6. Résultats obtenus des trois modèles de classification 35
7. Discussion des résultats 37
   - 7.1 Analyse comparative des performances 37
   - 7.2 Interprétation des courbes de perte 37
   - 7.3 Synthèse critique 38
   - 7.4 Exploitation des modèles pour des nouveaux textes 38

Conclusion 40

Conclusion générale 41

Liste des abréviations

AraBERT : Arabic Bidirectional Encoder Representations from Transformers

BR : Binary Relevance (Pertinence binaire)

BiLSTM : Bidirectional Long Short-Term Memory

CC : Classifier Chains (Chaînes de classificateurs)

CNN : Convolutional Neural Network (Réseau de neurones convolutif)

GRU : Gated Recurrent Unit

LP : Label Powerset (Ensemble d'étiquettes)

LSTM : Long Short-Term Memory

NLP : Natural Language Processing (le Traitement Automatique du Langage Naturel)

NLTK : Natural Language Toolkit

RNN : Recurrent Neural Network (Réseau de neurones récurrents)

SVM : Support Vector Machine

Liste des figures

Figure 1 : Multi- étiquette vs Multi-class 12
Figure 2 : Binary Relevance BR 12
Figure 3 : Label Powerset LP 13
Figure 4 : Classifier chains CC 13
Figure 5 : Couches (groupement de neurones) 17
Figure 6 : les fonctions d'activation 18
Figure 7 : Architecture RNN 19
Figure 8 : Approche traditionnelle vs. Approche de Transfert Learning 21
Figure 9 : Architecture du système proposé. 25
Figure 10 : Structure d'une base de données multi-étiquetée. 26
Figure 11 : Exemple d'un fichier texte 27
Figure 12: Les bi-grammes les plus fréquents dans la base de données 27
Figure 13: Texte avant / après Prétraitement 28
Figure 14: kaggle 30
Figure 15: Bibliothèques Python utilisées 32
Figure 16 : Venn Diagram for Category Overlap 33
Figure 17 : Matrice de cooccurrence des étiquettes 34
Figure 18 : perte de données selon les époques pour la méthode ARABERT 35
Figure 19 : perte de données selon les époques pour la méthode BiLSTM 36
Figure 20 : perte de données selon les époques pour la méthode CNN 36
Figure 21: les résultats de prédiction obtenus en utilisant le modèle AraBert 39
Figure 22: les résultats de prédiction obtenus en utilisant le modèle BiLSTM 39
Figure 23: les résultats de prédiction obtenus en utilisant le modèle CNN 40

Liste des tableaux

Tableau 1 : Table de matériels 30

Tableau 2 : Comparaison des performances des trois modèles 35

Introduction Générale

1. Contexte Générale :

La langue arabe est l'une des langues les plus parlées au monde, avec plus de 400 millions de locuteurs natifs répartis dans 22 pays. Elle est également une langue officielle des Nations Unies et joue un rôle fondamental dans la culture, l'éducation et la communication à l'échelle internationale.

Grâce à son alphabet unique et à sa richesse morphologique, l'arabe présente des défis spécifiques dans le domaine du traitement automatique du langage. Sa structure grammaticale complexe et la diversité de ses dialectes rendent l'analyse et la classification des textes plus difficiles que pour les langues latines.

2. Problématique et Objectifs

La classification des textes est une tâche clé du traitement du langage naturel qui permet d'organiser et d'exploiter efficacement l'information contenue dans des documents numériques. Avec la croissance exponentielle des données textuelles disponibles sur internet et dans les bases de données, il devient essentiel de développer des méthodes robustes et précises pour automatiser cette tâche. Cependant, lorsqu'il s'agit de la langue arabe, plusieurs défis rendent cette mission particulièrement complexe.

La langue arabe est caractérisée par une richesse morphologique et une complexité grammaticale qui influencent directement les performances des modèles de classification. De plus, l'absence de voyelles courtes dans l'écriture introduit des ambiguïtés qui rendent l'analyse automatique plus difficile. La diversité des dialectes arabes pose également un problème majeur, car ils diffèrent considérablement de l'arabe standard moderne, tant sur le plan lexical que syntaxique. En outre, contrairement à d'autres langues comme l'anglais, les ressources linguistiques en arabe, telles que les corpus annotés et les lexiques spécialisés, restent limitées, ce qui complique davantage l'apprentissage automatique des modèles.

Une autre limitation des recherches existantes est qu'elles se concentrent principalement sur la classification mono-label, où chaque document est associé à une seule catégorie. Or, dans de nombreux cas d'application, un texte peut appartenir à plusieurs catégories simultanément, ce qui relève du cadre de la classification multi-label (multi-étiquette). Ce type de classification, bien que plus représentatif de nombreux scénarios réels, a été peu exploré dans le contexte de la langue arabe.

Ce projet vise à étudier et comparer les performances des modèles de deep learning appliqués à la classification multi-étiquette des textes arabes. L'objectif est d'analyser l'impact des différentes techniques de représentation des textes et d'évaluer l'efficacité des architectures neuronales les plus avancées. En développant et en testant ces modèles sur des données en arabe, ce travail contribuera à améliorer la compréhension et la précision des systèmes de classification automatique dans cette langue.

3. Organisation du Mémoire

Ce mémoire est structuré comme suit :

- **Chapitre 1 : La catégorisation des documents arabe.** Ce chapitre présente les fondements de la classification automatique de textes, en mettant l'accent sur les spécificités de la langue arabe. Il aborde les principales méthodes de catégorisation, les défis linguistiques posés par l'arabe, ainsi que les approches de prétraitement, de représentation des textes et d'évaluation adaptées à cette langue.
- **Chapitre 2 : L'apprentissage profond pour la classification des textes**. Ce chapitre introduit les concepts clés de l'apprentissage profond appliqué au traitement automatique du langage naturel (TAL). Il décrit en particulier les architectures de réseaux neuronaux utilisées pour la classification de textes, telles que CNN, BiLSTM et les modèles préentraînés comme AraBERT, tout en mettant en évidence leurs avantages respectifs.
- **Chapitre 3** : **Modélisation et Implémentation.** Ce chapitre présente les étapes pratiques du projet de classification multi-étiquette. Il comprend la description du corpus utilisé, la mise en œuvre des différents modèles, l'évaluation des performances à l'aide de métriques standard, ainsi qu'une comparaison détaillée des résultats obtenus.

Chapitre 1 : La catégorisation des documents arabe

1. Introduction

La catégorisation des documents est essentielle pour organiser l'information et faciliter son accès. La langue arabe présente des défis particuliers en raison de sa complexité linguistique et de ses nombreuses variations.

Ce chapitre se propose d'explorer les principales méthodes utilisées pour la classification des textes arabes, en mettant en évidence les défis spécifiques liés à cette langue, les approches adoptées, ainsi que les techniques d'évaluation pertinentes.

2. Définition de la catégorisation des textes

La catégorisation des textes est une tâche centrale du traitement automatique du langage naturel (NLP), qui consiste à associer un ou plusieurs labels pertinents à un document textuel, à partir d'un ensemble de catégories prédéfinies.

Elle vise à automatiser l'organisation thématique des contenus textuels selon leur sémantique, en s'appuyant sur des algorithmes d'apprentissage automatique supervisé ou non supervisé. Cette tâche est essentielle dans divers domaines tels que la veille informationnelle, la gestion documentaire, la détection de sujets, ou encore l'analyse d'opinion. [1]

3. Difficultés et défis de la catégorisation des textes écrits en langue arabe

3.1 Complexité morphologique

L'arabe est une langue dotée d'une grande morphologie. Cela signifie qu'un mot peut devenir (ou changer de forme) selon sa fonction grammaticale (genre, nombre, temps, etc.). Par exemple : Le mot « **كَتَبَ** » (il a écrit) peut devenir « **كَتَبُوا** » (ils ont écrit), « **كَتَبْنَا** » (nous avons écrit), « **يَكْتُبُونَ** » (ils écrivent)… etc.

Cette richesse se traduit par de nombreuses variantes d'un même mot, ce qui augmente la difficulté pour un algorithme d'évaluer si tous ces mots sont liés à la même racine (ك-ت-ب).

3.2. Absence de Voyelles Courtes (Diacritiques)

Dans les textes arabes traditionnels (journaux, réseaux sociaux, etc.) – Les passages courts (fatha, damma, kasra) ne sont pas écrits. Cela crée de la confusion :

Le mot **« علم »** peut se lire :

- **عِلْم**= science
- **عَلَم**= drapeau
- **عَلِمَ**= il a su

Il faut donc prendre le sens exact du contexte comme entrée du modèle (ce qui peut être très difficile pour les modèles simples ou mal codés).

3.3 Manque de corpus annotés et de ressources

Les modèles de classification (deep learning plus spécifiquement) nécessitent des annotations : des textes étiquetés selon leurs catégories (politique, sport, religion... ou simplement plusieurs à la fois pour le modèle multi-étiquettes).

En arabe : Il existe peu de corpus publics et bien annotés. Le corpus disponible est généralement petit, déséquilibré (dans certaines classes, les sous-représentations sont évitées) ou non standardisé. Cela rend difficile l'entraînement de modèles performants.

3.4 Racines et dérivations

L'une des propriétés linguistiques les plus importantes de la langue arabe est le système de racines et de dérivés qui est construit sur un modèle morphologique non concaténatif dans lequel les mots sont constitués de racines trilittes (phares quadratiques), à partir desquelles de nombreux mots sont dérivés en appliquant diverses règles (modèles morphologiques). Exemple de racine :

Prenons la racine **ك-ت-ب**(k-t-b), qui exprime généralement l'idée d'**écriture**. À partir de cette racine, on peut dériver :

- **كَتَبَ**(kataba) : il a écrit

- كِتَاب(kitāb) : un livre
- مَكْتَب(maktab) : un bureau
- مَكْتَبَة(maktaba) : une bibliothèque
- يَكْتُبُونَ(yaktubūn) : ils écrivent
- مَكْتُوب(maktūb) : écrit (participe passé)

Ils ont des formes très différentes, mais sont sémantiquement liés à la même idée. Autrement dit, un modèle classique de classification ne reconnaîtrait probablement pas que ces mots partagent une racine, et donc un sens similaire [2].

4. Approches de catégorisation de documents pour la langue arabe

4.1 Approches classique

La catégorisation des documents dans les corpus arabes a historiquement reposé sur des approches traditionnelles, notamment [3] :

- **Classification manuelle**

Cette approche repose sur l'intervention d'experts qui attribuent manuellement des étiquettes aux textes.

- **Lexiques prédéfinis**

Utilisation de dictionnaires ou de listes de mots-clés associés à des catégories spécifiques. Par exemple, une classification des articles de presse basée sur la présence de mots spécifiques (ex. "football", "match" → catégorie "Sport").

- **Approche taxonomique**

Organisation des concepts sous forme de hiérarchie ou d'ontologie. Par exemple, une classification des documents médicaux basée sur la hiérarchie des maladies et des symptômes.

4.2 Approches basées sur le traitement automatique de langues naturelles

Les avancées technologiques ont permis l'émergence de nouvelles approches basées sur le Traitement Automatique du Langage Naturel (TALN) pour la classification thématique des corpus arabes. Ces méthodes modernes tirent parti de techniques avancées afin d'analyser et d'interpréter les textes de manière automatisée [4].

4.2.1 Approches basées sur l'apprentissage automatique (Machine Learning)

Dans cette partie, nous allons présenter les principaux modèles et algorithmes utilisés dans le cadre de la machine learning traditionnel.

- **Support Vector Machines (SVM) :**

Les SVM, ou machine à vecteurs de support, sont un des modèles les plus robustes et polyvalents en ML classique. Nés dans les années 1990, les SVM ont gagné en popularité grâce à leur capacité à effectuer à la fois des classifications linéaires et non linéaires avec une grande précision. Ce modèle est particulièrement réputé pour sa capacité à gérer des espaces de caractéristiques de haute dimension et pour sa flexibilité dans le choix des fonctions de noyau, ce qui lui permet de s'adapter à une grande variété de données.

- **Random forest :**

L'algorithme Random Forest appelé foret aléatoire constitue une méthode avancée d'apprentissage automatique, reconnue pour sa puissance et sa robustesse. Son principe fondamental repose sur la création d'une "forêt" d'arbres de décision, chacun construit de manière indépendante en utilisant des échantillons aléatoires des données d'entraînement et des caractéristiques spécifiques. Cette diversité dans la construction des arbres contribue à réduire le surajustement, permettant au modèle de généraliser efficacement sur de nouvelles données.

- **Naïve bayes :**

Le modèle Naïve Bayes est un algorithme d'apprentissage automatique basé sur le théorème de Bayes. Son approche repose sur la supposition "naïve" que les caractéristiques utilisées pour la classification sont conditionnellement indépendantes, ce qui simplifie considérablement le calcul des probabilités.

Le modèle Naïve Bayes présente plusieurs avantages, notamment sa simplicité et sa rapidité de mise en œuvre, ainsi que sa compétitivité dans de nombreuses tâches de classification, notamment dans les domaines de la classification de texte et de la catégorisation de documents. De plus, il est efficace avec des jeux de données de petite taille et moins sensible au surajustement par rapport à des modèles plus complexes, ce qui le rend robuste dans certaines situations. Cependant, le modèle comporte des limitations telles que son hypothèse d'indépendance conditionnelle pouvant être trop simpliste dans des situations de fortes corrélations entre les caractéristiques, des estimations de

probabilité inexactes, l'assomption d'une distribution normale des données, et un biais potentiel vers la classe majoritaire en présence de données déséquilibrées.

4.2.2 Approches basées sur l'apprentissage profond (Deep Learning)

- **Réseaux de neurones convolutifs (CNN)**

Un Convolutional Neural Network (CNN), ou réseau de neurones convolutif en français, est une architecture spécifique de réseaux de neurones profonds utilisée en apprentissage automatique et notamment en **Deep Learning.**

Ces réseaux de neurones ont la capacité d'apprendre à extraire des caractéristiques locales, c'est-à-dire des structures présentes de manière répétée à travers l'image. Au cœur du réseau neuronal convolutif, différentes couches, comme la **couche de convolution,** jouent un rôle central. La convolution permet aux CNNs d'analyser et de comprendre les structures visuelles complexes présentes dans les données d'entrée. Les réseaux de neurones convolutifs sont l'un des modèles les plus réputés pour la classification d'images en raison de leurs performances exceptionnelles.

- **Réseaux de neurones récurrents (RNN)**

Les RNN, ou Réseaux de Neurones Récurrents, sont une composante essentielle dans le paysage de l'intelligence artificielle, en particulier pour le traitement de données séquentielles telles que le texte ou les séries temporelles. Ces réseaux se distinguent des modèles traditionnels par leur capacité à maintenir un 'état' ou une mémoire des entrées précédentes, ce qui les rend particulièrement efficaces pour comprendre des séquences où le contexte et l'ordre sont importants. Les RNN ont joué un rôle clé dans le progrès de domaines tels que le traitement du langage naturel et l'analyse de séries temporelles. Leur aptitude à traiter des séquences de données leur permet de 'comprendre' le flux d'informations d'une manière sophistiquée, ouvrant la voie à des applications avancées telles que les assistants vocaux intelligents, les systèmes de recommandation personnalisés et les prévisions météorologiques précises.

Les réseaux de neurones récurrents (RNN) sont bien adaptés au traitement de données séquentielles grâce à leur structure en boucle, où la sortie d'une couche devient une partie de l'entrée de la même couche au moment suivant, créant ainsi une mémoire séquentielle. Pour surmonter les limitations des RNN standard dans la gestion des dépendances temporelles à long terme, des variantes avancées telles que les LSTM (Long

Short-Term Memory) et les GRU (Gated Recurrent Units) ont été développées. Cependant, l'entraînement efficace des RNN peut être difficile en raison de problèmes tels que la disparition ou l'explosion du gradient, ce qui a stimulé la recherche de techniques d'optimisation pour ces réseaux. Comprendre ces architectures, avec leurs forces et leurs limites, est crucial pour apprécier leur rôle dans le traitement séquentiel des données et leur application potentielle dans divers domaines.

5. Processus de catégorisation des textes

La catégorisation automatique des textes suit un processus structuré, composé de plusieurs étapes interdépendantes, visant à transformer un texte brut en une représentation exploitable pour la classification. Ce processus débute généralement par le **prétraitement** des textes, une phase essentielle qui permet de nettoyer et normaliser les données linguistiques. Ensuite, les documents sont convertis en **vecteurs numériques** à travers des méthodes de **représentation**, telles que TF-IDF ou Word2Vec, afin de capturer les informations sémantiques et contextuelles nécessaires à l'analyse. Une fois les données représentées sous une forme adéquate, elles sont soumises à des **algorithmes de classification**, qui apprennent à associer les textes à une ou plusieurs catégories en fonction de caractéristiques extraites. Enfin, les performances des modèles générés sont évaluées à l'aide de **métriques adaptées**, particulièrement dans le cadre de la classification multi-étiquette, pour garantir la fiabilité et la précision du système. Les sections suivantes détaillent chacune de ces étapes. [5]

5.1 Prétraitement des textes

Avant d'appliquer un algorithme de classification, les textes doivent être nettoyés et normalisés pour garantir une meilleure performance des modèles. Les étapes typiques incluent :

- **Nettoyage du texte** : Suppression des caractères spéciaux, ponctuation, chiffres, URL, et balises HTML.
- **Tokenization** : le texte est décomposé en formes de texte plus petites et plus simples pour une classification facile.
- **Normalisation** : Tout le texte d'un document doit être au même niveau de compréhension. Voici quelques formes de normalisation :

- Respecter les normes grammaticales ou structurelles dans l'ensemble du texte, par exemple en supprimant les espaces ou les signes de ponctuation. Ou encore, conserver les minuscules dans tout le texte.
- Supprimer les préfixes et les suffixes des mots et les ramener à leur racine.
- Suppression des mots vides tels que « et », « est », « le » et autres qui n'ajoutent pas de valeur au texte.
- **Suppression des mots vides (Stopwords Removal)** : Élimination des mots sans signification forte (ex. "هذا" ,"الذي").
- **Stemming et Lemmatisation** : Réduction des mots à leur racine ou forme canonique (ex. "كتب" → "يكتب").

5.2 Représentation des Textes

Une fois les textes prédéfinis transformés en une forme d'entrée donnée, utilisable par les algorithmes de classification, cette étape est essentielle car la qualité de la représentation affecte directement les performances du modèle. Il existe clairement deux approches principales [5] :

- **Bag of Words (BoW)** :

Une méthode dans laquelle le texte est représenté par un vecteur indiquant la fréquence de chaque mot d'un groupe verbal donné. Bien que cette méthode soit généralement simple, elle ignore l'ordre des mots et le contexte sémantique de chaque mot, ce qui peut entraver sa capacité à saisir le sens global d'un groupe [5].

- **Word Embeddings**

Contrairement aux incorporations BoW (par exemple, les modèles de type Word2Vec, GloVe ou BERT), elles permettent de représenter les mots sous forme de vecteurs denses dans un espace continu de faible dimension.

Ces représentations capturent les relations sémantiques et syntaxiques entre les mots et permettent ainsi une meilleure expression. En ce qui concerne l'analyse de textes arabes, des modèles génériques tels qu'AraBERT sont particulièrement utiles pour produire des incorporations contextuelles adaptées à la langue [5].

Nous allons présenter avec plus en détail la phase de représentation des textes dans la section suivante.

5.3 Classification des textes

5.3.1 Choix des Classificateurs

Nos expérimentations s'est appuyée sur trois modèles de Deep Learning : **CNN**, **BiLSTM** et **AraBERT**, en raison de leur capacité à traiter efficacement les textes en langue arabe.

Ces modèles permettent d'extraire des représentations contextuelles riches et sont particulièrement adaptés aux tâches complexes telles que la classification multi-étiquette. Ces modèles seront présentés plus en détail dans le chapitre suivant [5].

5.3.2 Evaluation des performances des classifieurs

Après l'entraînement des modèles, il est crucial d'évaluer leur performance en utilisant des métriques adaptées à la classification. Les métriques d'évaluation que nous avons adoptées sont présentées en détails dans la section 9. Métriques d'évaluation pour la classification multi-étiquette.

5.4 Application de classifier choisis pour la catégorisation des nouveaux textes

Une fois le modèle de classification entraîné et évalué, il peut être utilisé pour classer de nouveaux textes

6. Représentation des textes

6.1 TF-IDF

Le **TF-IDF** (de l'anglais *term frequency-inverse document frequency*) est une méthode de pondération souvent utilisée en recherche d'information et en particulier dans la fouille de texte. Cette mesure statistique permet d'évaluer l'importance d'un terme contenu dans un document, relativement à une collection ou un corpus [6].

Etant donné un corpus de *N* documents, La formule de TF-IDF pour un mot *w* dans un document d $d \in N$ est donnée par :

$$TF - IDF(w, d) = TF(w, d) * IDF(w)$$

Avec

$$TF(w, d) = \frac{nombre\ de\ fois\ que\ le\ terme\ \boldsymbol{t}\ apparaît\ dans\ le\ document}{\text{nombre total de mots dans le document d}}$$

$$IDF(w) = \log(\frac{nombre\ total\ de\ documents\ dans\ le\ corpus\ (N)}{\text{nombre de documents contenant le terme t}})$$

6.2 Word2Vec

Word2vec est une méthode basée sur l'apprentissage profond qui apprend des représentations denses (embeddings) pour les mots en fonction de leur contexte. Développé par Mikolov et al. (2013), Word2Vec repose sur deux architectures principales [7] :

- **CBOW (Continuous Bag of Words)** : prédit un mot cible à partir de son contexte.
- **Skip-Gram** : prédit les mots du contexte à partir d'un mot cible.

6.3 FastText

FastText est une technique de vectorisation des mots développée par Facebook AI, qui représente chaque mot comme une combinaison de sous-mots, permettant une meilleure gestion des mots rares ou inconnus.

7. Classification

La classification est un aspect clé de l'apprentissage automatique, consistant à assigner une ou plusieurs étiquettes (labels) à un texte. On distingue plusieurs types de classification, notamment la classification multi-classe et la classification multi- étiquette. [8]

7.1 Classification Multi-classe

La classification multi classe est un mécanisme d'apprentissage supervisé par lequel un modèle efficace classe une ressource dans une seule catégorie parmi plusieurs classes possibles**.** Dans une classification multi-classe, un exemple appartient à **une seule classe parmi plusieurs possibles.**

7.2 Classification Multi- étiquette

La classification multi- étiquette dans le texte est une technique d'apprentissage automatique qui consiste à attribuer plusieurs label ou catégories à un document ou à un morceau de texte. Contrairement à la classification traditionnelle qui attribue un seul label à chaque document, la classification multi-label permet de gérer des situations où un texte peut être associé à plusieurs catégories simultanément. Par exemple, dans le domaine de la classification des documents, un article peut être considéré comme appartenant aux

catégories "social", "politique" et "religion". La classification multi-label offre donc une approche plus flexible et plus précise pour organiser et catégoriser un large volume de texte en fonction de plusieurs critères. Dans une classification multi-label, un exemple peut appartenir à **plusieurs classes simultanément** [8].

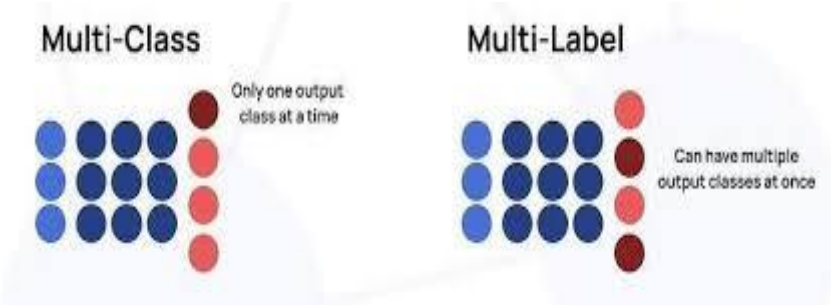

Figure 1 : Multi- étiquette vs Multi-class

8. Les approches de transformation du problème de classification multi-étiquette

8.1 Pertinence binaire (Binary Relevance BR)

La méthode de la pertinence binaire est un algorithme élémentaire qui décompose (ou bien qui transforme) un problème d'apprentissage à étiquettes multiples en un ensemble de problèmes de classification binaires indépendants [9]. La figure suivante illustre la méthode de pertinence binaire.

X	Y_1	Y_2	Y_3	Y_4
$X^{(1)}$	0	0	0	1
$X^{(2)}$	1	0	0	0
$X^{(3)}$	1	0	0	1
$X^{(4)}$	0	1	0	0
$X^{(5)}$	0	1	1	0

X	Y_1
$X^{(1)}$	0
$X^{(2)}$	1
$X^{(3)}$	1
$X^{(4)}$	0
$X^{(5)}$	0

X	Y_2
$X^{(1)}$	0
$X^{(2)}$	0
$X^{(3)}$	0
$X^{(4)}$	1
$X^{(5)}$	1

X	Y_3
$X^{(1)}$	0
$X^{(2)}$	0
$X^{(3)}$	0
$X^{(4)}$	0
$X^{(5)}$	1

Figure 2 : Binary Relevance BR

8.2 Ensemble de puissances d'étiquettes (Label Powerset LP)

Comme il est présenté dans la Figure 3, la méthode de puissance d'étiquettes (LP) transforme le problème d'apprentissage multi-label en un seul problème d'apprentissage mono label à plusieurs classes [9].

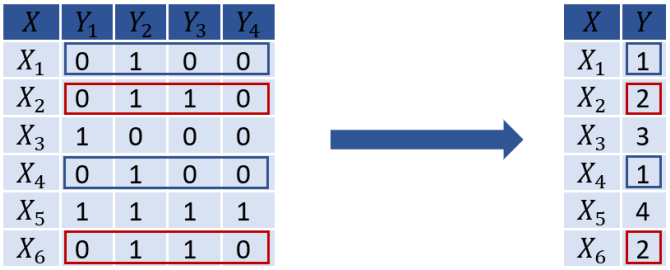

Figure 3 : Label Powerset LP

8.3 Chaîne de classificateurs (Classifier chains CC)

La méthode de chaine de classification est une amélioration de la méthode BR qui transforme également le problème d'apprentissage multi-label en q problèmes de classification ou de régression mono-label [9]. La figure suivante donne un exemple de cette méthode.

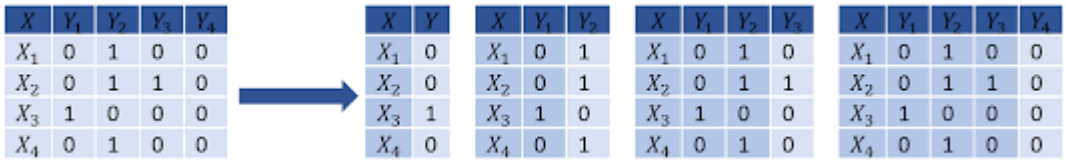

Figure 4 : Classifier chains CC

9. Métriques d'évaluation pour la classification multi-étiquette

Dans la classification multi-label, chaque document peut appartenir à plusieurs catégories simultanément. Évaluer la performance d'un modèle dans ce contexte nécessite des métriques spécifiques adaptées à cette complexité. Voici les principales métriques utilisées [10] :

9.1 Coût de Hamming (Hamming Loss)

Ce critère évalue combien de labels sont mal classés. Mesure le pourcentage de mauvaises prédictions faites par le modèle, en termes de l'ensemble des étiquettes possibles. Voici l'équation pour calculer le Hamming

$$\boldsymbol{Hamming\ Loss} = \frac{1}{\mathrm{N} \times L} \times \sum_{i=1}^{N} \sum_{j=1}^{L} I(\mathrm{yi,j} \neq \mathrm{y'\,i,j})$$

Où : N représente le nombre d'échantillons (documents, observations), L est le nombre total de labels (classes) possibles, $y_{i,\ j}$ représente la classe réelle (true label) pour l'échantillon i et l'étiquette j et $y'_{i,j}$ est la classe prédite pour l'échantillon i et l'étiquette j.

9.2 Accuracy

L'exactitude, en anglais *accuracy*, représente le pourcentage de bonne prédiction, est une mesure permettant d'évaluer les modèles de classification basée sur la matrice de confusion, elle mesure le taux de prédictions correcte sur l'ensemble des individus. Voici l'équation pour calculer :

$$\text{Accuracy} = \frac{\text{nombre de predictions correctes}}{\text{nombre total de predictions}}$$

9.3 Précision

La précision également appelée valeur prédictive positive (taux de prédictions correctes), c'est le rapport entre les observations positives (prédite) et le nombre total d'observations positives prédites. Il mesure la capacité du modèle à ne pas faire d'erreur lors de prédiction positives.

$$Precision = \frac{\text{nombres de vrais positifs}}{\text{nombres de vrais positifs } + \text{ nombres de faux positifs}}$$

9.4 Rappel

Rappel, en anglais *recall*, est défini comme le nombre d'instances de la classe positive qui étaient correctement prédit. Ceci est également connu sous le nom de couverture ou de sensibilité et peut être représenté par la formule :

$$Rappel = \frac{\text{nombres de vrais positifs}}{\text{nombres de vrais positifs } + \text{ nombres de faux } négatif}$$

9.5 F-score

Le F1-score est une autre mesure de précision qui est calculée en prenant la moyenne harmonique de la précision et du rappel et elle est représentée comme suit :

$$F - mesure = 2 \times \frac{\text{precesion } \times \text{ rappel}}{\text{precesion } + \text{ rappel}}$$

Conclusion

La catégorisation des documents en arabe est une tâche essentielle mais complexe en raison des particularités de la langue. Différentes approches, classiques et basées sur l'apprentissage automatique, ont été développées pour améliorer la précision de la classification. Grâce aux avancées en intelligence artificielle, la performance des modèles continue de s'améliorer, mais des défis subsistent, nécessitant encore des recherches et des optimisations pour mieux traiter la langue arabe.

Chapitre 2 : L'apprentissage profond pour la classification des textes

1. Introduction

L'apprentissage profond (**Deep Learning**) est une approche puissante qui permet d'automatiser l'extraction de caractéristiques à partir des textes et d'améliorer la classification des documents. Contrairement aux méthodes traditionnelles, il utilise des réseaux de neurones artificiels (RNA) pour apprendre des représentations complexes des données textuelles.

Dans ce chapitre, nous présenterons les réseaux de neurones, leurs composants essentiels, ainsi que les principales architectures utilisées pour la classification des textes.

2. Réseaux de Neurones Artificiels (RNA)

Les réseaux de neurones artificiels sont des modèles informatiques, implémentés en matériel ou en logiciel, conçus pour s'inspirer du fonctionnement des neurones biologiques du cerveau humain. Ils font partie des technologies d'apprentissage profond (**Deep Learning**), une sous-catégorie de l'intelligence artificielle [11].

3. Les principaux composants du réseau de neurones

Un **réseau de neurones artificiel** est constitué de plusieurs éléments fondamentaux qui travaillent ensemble pour traiter les données et effectuer des prédictions [12].

3.1 Neurones

Le neurone est une unité qui reçoit l'information, procède à des calculs simples, et la transmet à une autre unité.

3.2 Couches (groupement de neurones)

Comme il est illustré dans la Figure 5, il existe trois types de couches dans un réseau de neurones artificiel :

- **Couche d'entrée**

  Cette couche est celle qui réceptionne les informations brutes. Par comparaison, nous pouvons considérer que les nerfs optiques perçoivent une image avant de l'analyser.

- **Couches intermédiaires cachées**

A donnée brute va transiter par plusieurs couches qui chacune va opérer une analyse partielle de l'information avant de la transmettre à la suivante.

- **Couche de sortie**

La toute dernière couche produit le résultat final, soit l'information analysée.

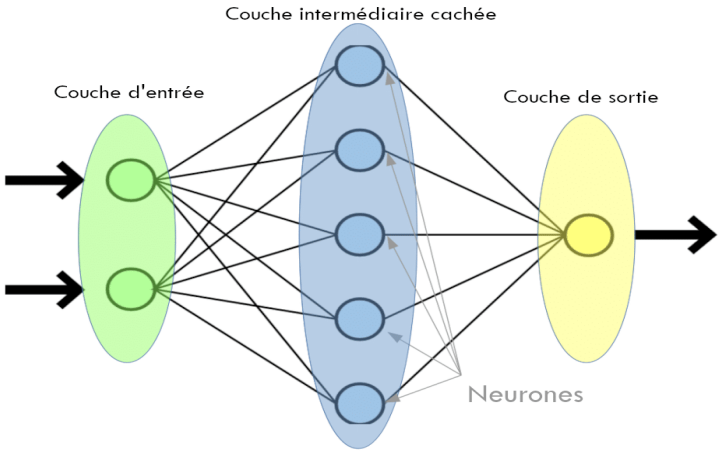

Figure 5 : Couches (groupement de neurones)

3.3 Poids et biais

Les poids et biais sont des variables du modèle qui sont mises à jour afin d'améliorer la précision du réseau. Un poids est appliqué à l'entrée de chaque neurone pour calculer une valeur de sortie. Les réseaux de neurones mettent à jour ces poids de manière continue.

Il existe donc une boucle de rétroaction (ou rétropropagation) mise en œuvre dans la plupart des réseaux de neurones.

Les biais sont aussi des valeurs numériques, ajoutées après l'application des poids sur les entrées. Ainsi, les poids et les biais sont des valeurs apprises automatiquement par les réseaux de neurones [13].

3.4 Fonction d'activation

La fonction d'activation est une formule mathématique (algorithme) activée dans certaines circonstances. Lorsque les neurones calculent la somme pondérée des valeurs d'entrée + le biais, elles sont transmises à la fonction d'activation, qui vérifie si la valeur calculée est supérieure au seuil requis [13].

3.4.1 Types de fonctions d'activation

Les fonctions d'activation sont présentées dans la Figure 6. Il existe plusieurs fonctions d'activation telles que : *sigmoïde*, *tanh* et *ReLU*.

- **Sigmoïde :** produit une courbe en forme de S. Bien que de nature non linéaire, il ne tient toutefois pas compte des légères variations des entrées, ce qui entraîne des résultats similaires**.**
- **Fonctions de tangente hyperbolique (tanh) :** Il s'agit d'une fonction supérieure comparée à Sigmoïde. Cependant, elle rend moins bien compte des relations et elle est plus lente àconverger.
- **Unité linéaire rectifiée (ReLU) :** Cette fonction converge plus rapidement, optimise et produit la valeur souhaitée plus rapidement. C'est de loin la fonction d'activation la plus populaire utilisée dans les couches cachées.
- **Softmax :** utilisé dans la couche de sortie car il réduit les dimensions et peut représenter une distribution catégorique.

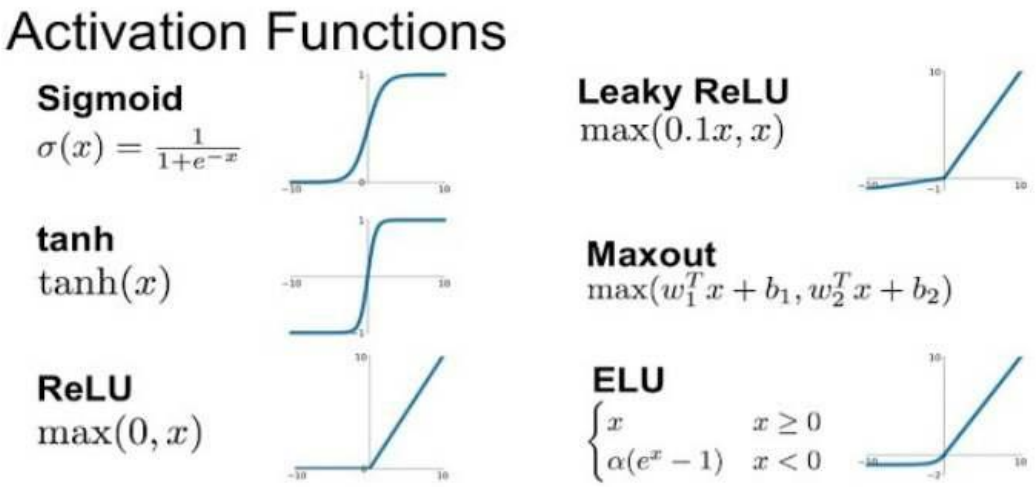

Figure 6 : les fonctions d'activation

4. Les méthodes d'apprentissage profond (deep learning) utilisée

L'apprentissage profond est une branche de l'apprentissage automatique qui repose sur des réseaux de neurones profonds pour analyser les données et en extraire des modèles. Dans le traitement automatique du langage naturel (TALN), les réseaux de neurones récurrents (RNNs) et leurs variantes améliorées comme LSTM et GRU sont largement utilisés pour traiter les données séquentielles, telles que les textes et les phrases.

Dans cette section, nous présentons les méthodes d'apprentissage profond utilisées dans notre étude.

4.1 Les réseaux de neurones récurrents (RNN)

Les réseaux de neurones récurrents RNN, sont un type de réseau de neurones largement utilisé dans le domaine de l'apprentissage en profondeur (Deep Learning). Les RNN utilisent les sorties précédentes comme entrées supplémentaires et **sont parfaitement adaptés au traitement de données séquentielles** [14]. Généralement, elles se présentent sous la forme suivante :

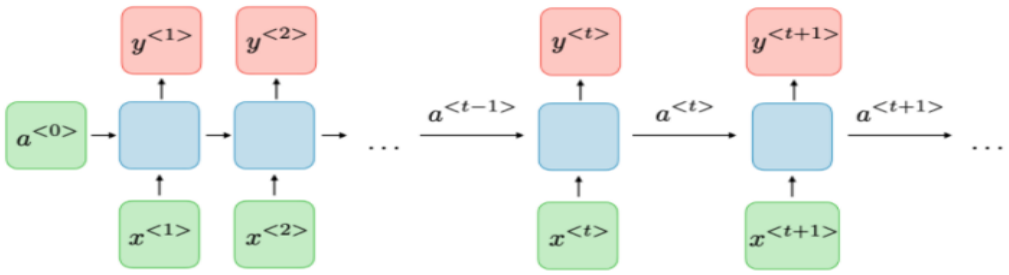

Figure 7 : Architecture RNN

4.2 Long-Short Term Memory (LSTM)

Les LSTM sont des dérivés de RNN. Ils peuvent apprendre et mémoriser des dépendances sur une longue durée. Les LSTM conservent ainsi les informations mémorisées sur le long terme. Ils sont particulièrement utiles pour prédire des séries chronologiques, car ils se rappellent des entrées précédentes. Outre ce cas d'utilisation, les LSTM sont également utilisés pour composer des notes de musique et reconnaître des voix.

Pour y remédier, les réseaux LSTM possèdent des « cellules » situées dans les couches cachées du réseau neuronal artificiel, qui possèdent trois portes : une porte d'entrée, une porte de sortie et une porte d'oubli. Ces portes contrôlent le flux d'informations nécessaires pour prévoir la sortie sur le réseau. Par exemple, si les pronoms de genre, tels que « elle », ont été répétés plusieurs fois dans les phrases précédentes, vous pouvez les exclure de l'état de la cellule [15].

4.2.1 Réseaux Bi-directionnels LSTM (BiLSTM)

Le BiLSTM constitue une évolution du modèle LSTM classique. Il traite l'information dans les deux directions temporelles (passée et future), ce qui permet d'extraire un contexte plus riche pour chaque mot ou élément de la séquence. Cette architecture est particulièrement performante dans les tâches de classification de textes où le contexte global joue un rôle crucial.

4.3 Réseaux de neurones convolutifs (CNN)

Les réseaux de neurones convolutives, Convolutional Neural Networks (CNN**),** sont un type de réseau de neurones profonds spécialement conçus pour extraire automatiquement les caractéristiques locales et spatiales à partir des données d'entrée, notamment les images. Leur architecture repose sur l'utilisation de couches de convolution, de fonctions d'activation non linéaires, de couches de pooling, et souvent de couches entièrement connectées à la fin.

Dans le cadre de la classification multi-étiquettes, où chaque instance peut appartenir simultanément à plusieurs classes, les CNN se révèlent particulièrement efficaces. Ils permettent de :

- Capturer des représentations hiérarchiques complexes des données ;
- Apprendre des caractéristiques discriminantes pertinentes pour distinguer plusieurs étiquettes en parallèle ;
- Être intégrés avec une fonction de perte adaptée (comme Binary Cross-Entropy) permettant la prédiction simultanée de plusieurs sorties indépendantes. Grâce à leur capacité à modéliser des relations non linéaires et à généraliser efficacement, les CNN sont largement utilisés dans des tâches telles que la **classification d'images multi-étiquettes**, la détection d'objets ou encore l'analyse de texte.

5. Transfer learning et modèle pré-entrainé

5.1 Transfer learning

Le transfert learning, ou **apprentissage par transfert** en français, est une technique d'apprentissage automatique. À travers cette méthode, les connaissances acquises à partir d'une tâche sont réutilisées pour améliorer les performances sur une tâche similaire. Cette

notion est de plus en plus en vogue et prend de la place dans le domaine de l'apprentissage profond [16]. La Figure 8 illustre la différence entre l'approche traditionnelle et l'approche de transfert learning.

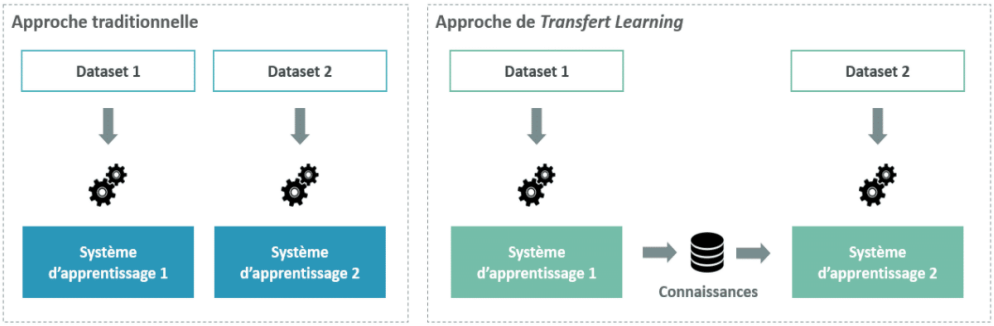

Figure 8 : Approche traditionnelle vs. Approche de Transfert Learning

5.1.1 Fonctionnement de l'apprentissage par transfert

Cette méthode consiste en trois principales étapes**.** Tout d'abord, l'utilisation d'un modèle pré-entraîné est requise. Ensuite, l'adaptation de ce modèle, et enfin, l'entraînement.

- **L'utilisation d'un modèle pré-entraîné**

Un modèle source pré-entraîné est requis. Il est choisi pour transférer ses connaissances précises au modèle cible. Ce modèle est généralement un réseau de neurones, ou plutôt un **empilement de couches de neurones**. Ce dernier a été entraîné grâce à une grande base de données. Et surtout, pour une mission bien définie, comme définir une image (chat ou chien, voiture ou camion, bus ou citerne, etc.). Les premières couches de neurones se concentrent sur les caractéristiques génériques, tandis que les couches plus profondes se focalisent sur la mission à accomplir (définir ce qu'il y a exactement sur l'image).

- **L'adaptation du modèle source pour créer le modèle cible**

L'adaptation est essentielle. Le modèle source peut présenter certaines caractéristiques différentes de celles du modèle cible. Il peut présenter des données d'entraînement différentes. Ainsi, il est indispensable de configurer et d'adapter ces

caractéristiques avant de transférer tous les acquis au modèle cible. Par exemple, le modèle source pourrait être capable de classer des milliers de catégories d'objets, alors que son homologue ne pourrait en classer que quelques-unes.

- **L'entraînement du modèle source pour obtenir le modèle cible**

Enfin, la préparation du modèle source est la dernière étape, et pas des moindres. Une fois le réglage du modèle source effectué, le modèle cible est conçu en utilisant le modèle source comme point de départ. Cette préparation permet d'obtenir de meilleures performances en un laps de temps réduit**.** De manière générale, la présence de modèles déjà bien installés est primordiale dans le domaine du deep learning.

5.1.2 Les principaux avantages du transfer learning

Le TL présente plusieurs avantages dans le domaine du Deep Learning et de l'IA :

- Il offre un démarrage rapide, car le sujet est pré-entraîné à l'accomplissement de la mission. Cela demande aussi moins de temps d'exercice, donc une utilisation des ressources informatiques plus efficace.
- Le TL permet aussi d'obtenir des modèles performants, même avec des datas limitées pour une mission cible.
- Il réduit aussi les coûts, car il est plus rapide.
- Il améliore les acquisitions générales utilisées pour des missions similaires.
- Il fournit également de meilleurs résultats que ceux qui ne sont pas entraînés.
- Il est très polyvalent et s'adapte rapidement à d'autres modèles.

5.1.3 Utilisation du transfert learning dans le traitement du langage naturel

Le transfer learning est largement utilisé dans le domaine du traitement du langage naturel. Il s'agit d'un domaine de l'intelligence artificielle basée sur l'interaction entre les ordinateurs et le langage humain naturel.

Il peut être utilisé pour :

- La classification de texte
- La reconnaissance d'entités
- La traduction automatique
- La génération de texte

- le développement d’assistance virtuelle
- l’analyse de texte, etc.

5.2 Modèle pré-entraîné

Un modèle pré-entraîné fait référence à une machine learning modèle qui a été préalablement formé sur un grand ensemble de données et qui est prêt à être peaufiné ou utilisé pour des tâches spécifiques. Ces modèles sont particulièrement utiles dans des domaines tels que les statistiques, l’analyse des données, et la science des données, où la formation d'un modèle à partir de zéro peut prendre du temps et nécessiter beaucoup de ressources. En exploitant des modèles pré-entraînés, les data scientists peuvent économiser du temps et des ressources de calcul tout en obtenant une grande précision dans leurs prévisions. [17]

5.2.1 Avantages de l'utilisation de modèles pré-entraînés

L’un des principaux avantages de l’utilisation de modèles pré-entraînés est la réduction du temps de formation. Étant donné que ces modèles ont déjà été formés sur des ensembles de données volumineux, ils sont dotés de fonctionnalités apprises qui peuvent être directement appliquées à de nouvelles tâches similaires.

Cela permet aux praticiens de se concentrer sur l’ajustement du modèle en fonction de leurs besoins spécifiques plutôt que de partir de zéro. De plus, les modèles pré-entraînés offrent souvent de meilleures performances, en particulier dans les scénarios avec des données limitées, car ils peuvent bien généraliser à partir des connaissances acquises lors de leur formation initiale. [17]

5.2.2 Applications courantes des modèles pré-entraînés

Les modèles pré-entraînés sont largement utilisés dans diverses applications, notamment le traitement du langage naturel (NLP), la vision par ordinateur et la reconnaissance vocale. En NLP, des modèles comme BERT et GPT-3 ont été pré-entraînés sur de vastes corpus de texte, ce qui leur permet de comprendre efficacement le contexte et la sémantique. [17]

5.2.3 Le modèle pré-entraîné AraBERT (**Arabic Bidirectional Encoder Representations from Transformers**)

AraBERT est un modèle de type Transformer pré-entraîné spécialement pour la langue arabe. Il est basé sur l'architecture BERT et permet de mieux comprendre le contexte des textes arabes, ce qui améliore les performances dans diverses tâches de traitement automatique du langage naturel (TALN), notamment la classification de texte.

Le modèle pré-entraîné AraBERT est développé par une équipe de chercheurs de l'Université américaine de Beyrouth et du laboratoire de recherche en intelligence artificielle "AUB MC Lab". Le modèle est construit sur l'architecture de Transformer et est entraîné sur un vaste corpus de texte en arabe. Il est capable de saisir les subtilités de la langue arabe, y compris les variations grammaticales et les mots conjugués, et de capturer les relations sémantiques et syntaxiques dans la langue en apprenant une représentation vectorielle dense pour chaque mot. De plus, le modèle est disponible gratuitement en open source, permettant ainsi un accès ouvert à tous.

Ce modèle est largement considéré comme la base de la plupart des résultats de pointe obtenus dans différentes tâches de NLP, dans plusieurs langues. Nous utilisons la configuration BERT, qui comprend 12 blocs d'encodeurs, 768 dimensions cachées, 12 têtes d'attention, une longueur de séquence maximale de 512, et un total d'environ 110 millions de paramètre [18].

Conclusion

L'apprentissage profond joue un rôle clé dans la classification des textes grâce aux réseaux de neurones artificiels et à leurs architectures avancées comme les RNNs, LSTM et BiLSTM. Des modèles plus récents basés sur les transformeurs, tels qu'AraBERT , ont également montré une performance remarquable, notamment pour le traitement du texte en langue arabe. Ces modèles permettent de mieux capturer le contexte et la structure des textes, améliorant ainsi la précision des classifications. Malgré certains défis, leur utilisation continue de progresser avec l'évolution des technologies.

Chapitre 3 : Modélisation et implémentation

1. Introduction

Dans ce travail, nous visons à développer un système intelligent de classification automatique de documents en langue arabe selon des thématiques multiples. Pour cela, nous avons conçu un programme performant capable de traiter et d'analyser de grands volumes de données textuelles. Le projet comprend plusieurs étapes, allant de la préparation et du prétraitement des données jusqu'à la construction de modèles de classification basés sur l'apprentissage par transfert à l'aide d'un modèle pré-entraîné. Nous présentons dans ce cadre l'environnement de travail ainsi que les outils utilisés, et nous concluons par une discussion sur les résultats obtenus et leur signification quant à l'efficacité du système proposé.

2. Architecture du système

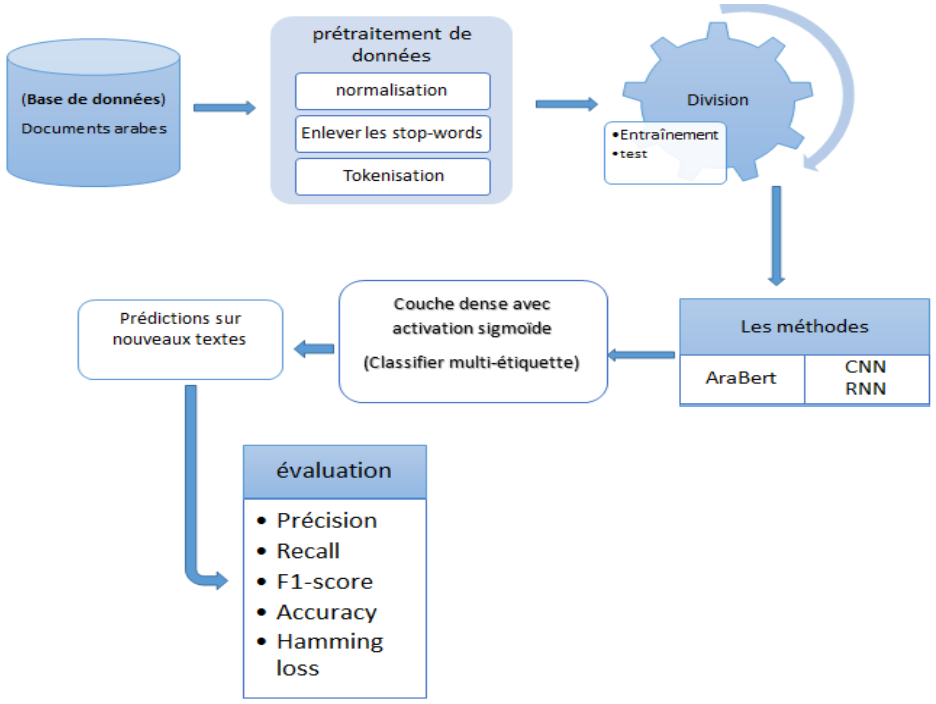

Figure 9 : Architecture du système proposé.

3. Fonctionnalités du système de catégorisation des documents

3.1 Description de base de donné

À notre connaissance, l'ensemble de données NADiA est la plus vaste source de données textuelles arabes, utilisable pour toute tâche liée au traitement du langage naturel (TALN), comme la classification de textes. Les données ont été collectées en récupérant les données des sites d'information « SkyNewsArabia » et « Masrawy » à l'aide de scripts Python optimisés pour chaque site. La base de données **NADiA1** se compose d'un ensemble de fichiers texte (35 416 fichiers) organisés dans un dossier principal, accompagnés d'un fichier `yaml` décrivant les différentes classes de catégorisation. Initialement, cette base contient **24 catégories** représentant divers domaines thématiques. Toutefois, dans le cadre de notre étude, nous avons choisi de **réduire le nombre de classes à trois uniquement**, à savoir : *Football*, *English League* et *Arabian Sports*. À la suite de cette sélection, le corpus a été réduit à 4 508 documents cette sélection vise à concentrer notre analyse sur un sous-ensemble cohérent et pertinent pour l'expérimentation, tout en facilitant l'évaluation comparative des modèles de classification multi-étiquette.

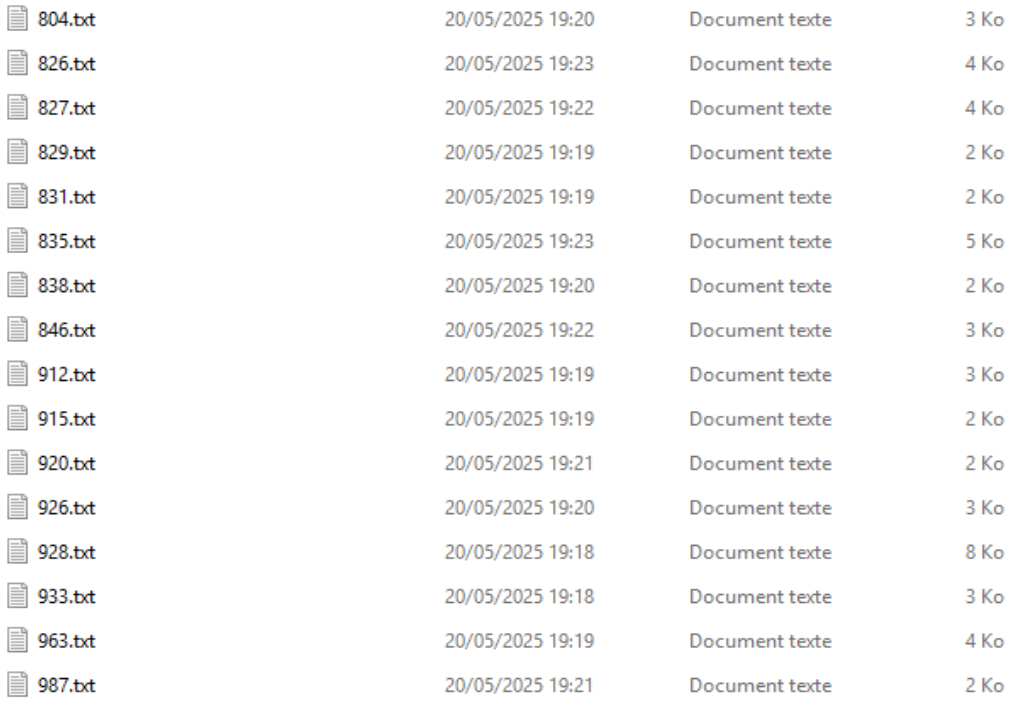

Figure 10 : Structure d'une base de données multi-étiquetée.

انتخابات إرهاب مكافحة_الإرهاب

وذكر صالح خلال مؤتمر صحفي عقده بالعاصمة اليمنية صنعاء أن ذهابه إلى أمريكا هو للابتعاد عن الأنظار وليس للعلاج لأنه بصحة جيدة حسب قوله. ونفى صالح أي خطة لبقائه في السلطة.

قال أنه سيغادر ليتيح الفرصة للحكومة المؤقتة التي تقودها المعارضة بالإعداد لانتخابات رئاسية مبكرة لاختيار رئيس جديد للبلاد في 21 فبراير المقبل.

وأكد صالح أنه سينسحب من العمل السياسي، وينزل إلى الشارع كجزء من المعارضة

وفي إشارة إلى العلاقة بين بلاده وجيرانه قال صالح إن عدم استقرار اليمن يعني عدم الاستقرار في المنطقة كلها وأن حماية الأمن والاستقرار في اليمن هو حماية للأمن في الدول المجاورة.

وقد فتح جنود من وحدات يقودها ابن صالح وابن أخيه النار على محتجين مما أدي إلي مقتل تسعة أشخاص إصابة نحو 150 أخرين. واستخدم الجنود الغاز المسيل للدموع وخراطيم المياه لتفريق متظاهرين اقتربوا من المجمع الرئاسي في صنعاء بعد مسيرة استمرت عدة أيام قادمين من مدينة تعز الجنوبية، وهم يرددون هتافات رافضة منح صالح الحصانة من المحاكمة.

Figure 11 : Exemple d'un fichier texte

3.1.1 Visualisation des bigrammes fréquents

Les bigrammes sont des paires de mots qui apparaissent successivement dans les textes. Leur visualisation permet d'identifier les associations de mots les plus fréquentes dans la base de données, ce qui peut aider à mieux comprendre les thématiques dominantes du corpus. La figure suivante illustre les bigrammes les plus fréquents extraits de la base de données textuelle.

	bi-gram	count
0	ريال مدريد	1577
1	في الدقيقة	1523
2	قبل أن	1073
3	في الدوري	1054
4	كأس العالم	1034
5	أبطال أوروبا	1006
6	هذا الموسم	866
7	دوري أبطال	863
8	في المركز	832
9	بعد أن	753
10	مليون يورو	689
11	على التوالي	684
12	ركلة جزاء	667
13	لكرة القدم	656
14	كرة القدم	639
15	رصيده إلى	626
16	مانشستر يونايتد	590
17	على ملعب	577
18	الموسم الماضي	563
19	إلى أن	542
20	من ركلة	525
21	الشوط الثاني	524
22	سان جرمان	500
23	في مباراة	495
24	مانشستر سيتي	493
25	في كأس	466
26	في المباراة	458
27	في دوري	444
28	من أجل	434
29	في العالم	431

Figure 12: Les bi-grammes les plus fréquents dans la base de données

3.2 Nettoyage et prétraitement de données

Avant de pouvoir exploiter les données extraites dans le cadre de notre projet, celles-ci doivent passer par une série d'étapes de prétraitement afin de les rendre adaptées à l'apprentissage automatique. Ce processus inclut généralement les opérations suivantes :

- Suppression des hashtags.
- Élimination des espaces superflus.
- Suppression des mots vides (stopwords) tels que « في », « ثم », etc.
- Suppression de la ponctuation, qu'elle soit arabe ou non arabe.
- Suppression des caractères spéciaux.
- Suppression des liens URL.
- Suppression des emojis.
- Suppression des mentions (par exemple : @nom).
- Élimination du phénomène de « التطويل », c'est-à-dire la répétition de lettres pour allonger un mot.
- Normalisation des ligatures : remplacement de « ء » par « ئ ».Normalisation de la hamza : remplacement de « ة » par « ه » et de « ي » final par « ى ».

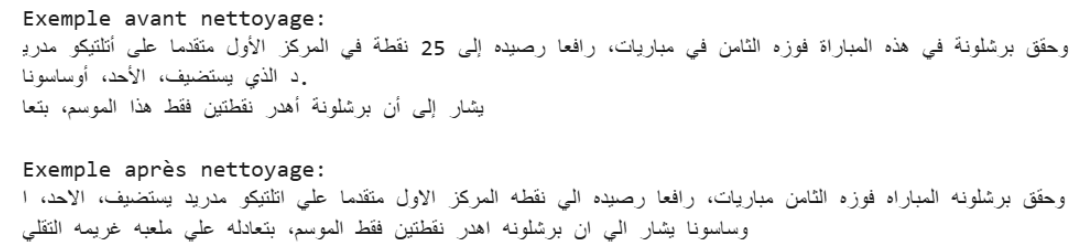

Figure 13: Texte avant / après Prétraitement

3.3 Catégorisation des documents

3.3.1 Répartition des données

Une fois les textes préparés et numérisés, l'étape suivante consiste à diviser l'ensemble des données en deux sous-ensembles : un ensemble d'entraînement et un ensemble de test. L'ensemble d'entraînement est utilisé pour ajuster les paramètres du modèle, tandis que l'ensemble de test sert à évaluer ses performances de manière objective. En général, la majorité des données est attribuée à l'entraînement, afin d'assurer un apprentissage efficace.

Il est essentiel que les données de test proviennent de la même distribution que celles d'entraînement, sans pour autant être utilisées durant la phase d'apprentissage, afin de garantir une évaluation impartiale. Cette séparation permet de mesurer la capacité du modèle à généraliser sur de nouvelles données jamais vues auparavant, ce qui constitue un critère fondamental de performance en apprentissage automatique.

3.3.2 Classification des documents

Notre système de catégorisation des documents repose sur des modèles de deep learning basés sur le transfert learning avec des modèles de langue pré-entraînés spécialisés pour l'arabe. Plus précisément, nous avons utilisé les modèles CNN, BiLSTM et AraBERT

3.4 L'évaluation

Dans l'évaluation des modèles de classification de texte à étiquettes multiples, le choix des métriques appropriées est essentiel pour évaluer les performances du modèle et sa capacité à généraliser sur de nouvelles données.

Notre système utilise des métriques telles que la précision (precision), le rappel (recall), la matrice de confusion, la F-mesure (F1-score) et l'exactitude (accuracy), chacune apportant une perspective complémentaire sur la qualité des prédictions du modèle.

4. Environnement de travail et outils d'implémentation

4.1 Matériel

Caractéristiques	Poste de travail
PC	Lenovo Thinkpad
Système d'exploitation	Windows 11 pro
Processeur	Intel(R) Core(TM) i5-10310U CPU @ 1.70GHz 2.21 GHz
RAM	8.00 g
Type de système	SE 64 bits

Tableau 1 : Table de matériels

4.2 Environnement de programmation

4.2.1 Langage de programmation

Nous avons choisi Python, pour implémenter notre système de catégorisation des documents. Python est un langage de programmation puissant et facile à apprendre. Il offre des structures de données de haut niveau et permet une approche simple mais efficace de la programmation orientée objet. Grâce à sa syntaxe élégante, son typage dynamique et son interprétation, Python est un langage idéal pour l'écriture de scripts et le développement rapide d'applications dans de nombreux domaines et sur la plupart des plateformes. [19]

4.2.2 Éditeur de code

- **Kaggle**

Kaggle est une plateforme communautaire en ligne destinée aux scientifiques des données et aux passionnés de l'apprentissage automatique. Kaggle permet aux utilisateurs de collaborer avec d'autres utilisateurs, de trouver et de publier des ensembles de données, d'utiliser des carnets de notes intégrés au GPU et de rivaliser avec d'autres scientifiques des données pour relever des défis dans le domaine de la science des données. L'objectif de cette plateforme en ligne est d'aider les professionnels et les apprenants à atteindre leurs objectifs dans leur parcours en science des données grâce aux puissants outils et ressources qu'elle met à leur disposition. [20]



Figure 14: kaggle

4.2.3 Librairies et bibliothèques Python intégrés

❖ Pandas

Pandas est une bibliothèque Python spécialisée dans la manipulation et l'analyse de données. Elle fournit des structures comme les DataFrames, qui permettent de gérer facilement des tableaux de données, de les filtrer, de les trier ou d'effectuer des opérations statistiques simples. Elle est largement utilisée dans les projets de science des données, de statistiques et de machine learning. [21]

❖ NumPy

NumPy est une bibliothèque Python utilisée pour effectuer des calculs numériques rapides sur de grandes quantités de données. Elle permet de manipuler des tableaux multidimensionnels appelés arrays et propose de nombreuses fonctions mathématiques et algébriques. [22]

❖ scikit-learn

Scikit-learn est une bibliothèque Python dédiée à l'apprentissage automatique. Elle propose des algorithmes pour la classification, la régression, le clustering et la réduction de dimensions. Elle est aussi utilisée pour évaluer les performances des modèles à l'aide de métriques standards comme la précision ou le score F1. [23]

❖ Matplotlib

Matplotlib est une bibliothèque de visualisation de données pour Python. Elle permet de créer des graphiques variés utiles pour explorer les données ou représenter les performances d'un modèle. [24]

❖ PyTorch

PyTorch est une bibliothèque open-source de deep learning développée par Facebook AI.Elle permet de créer, entraîner et évaluer des réseaux de neurones avec une grande flexibilité**.** [25]

❖ Transformers

Transformers est une bibliothèque développée par Hugging Face permettant d'utiliser facilement des modèles de traitement du langage naturel comme BERT ou RoBERTa. Elle

offre des outils pour la classification de texte, la génération de texte, la traduction, et plus encore. Elle est largement utilisée dans le traitement de la langue naturelle (NLP) moderne.[26]

❖ **NLTK (Natural Language Toolkit)**

NLTK est une bibliothèque Python spécialisée dans le **traitement automatique du langage naturel (TALN)**. Elle offre un ensemble complet d'outils pour analyser des textes en langage humain, comme la tokenisation, le stemming, la lemmatisation, l'étiquetage grammatical (POS tagging) et l'analyse syntaxique. NLTK fournit également l'accès à de nombreux corpus linguistiques (comme WordNet) et est largement utilisée dans la recherche et les projets d'analyse de texte. [27]

❖ TensorFlow

TensorFlow est une bibliothèque open-source développée par **Google** pour l'**apprentissage automatique** et le **deep learning**. Elle permet de construire, entraîner et déployer des réseaux de neurones complexes de manière efficace. TensorFlow prend en charge les calculs sur GPU/TPU, et propose une interface flexible pour créer des modèles de classification, de reconnaissance d'image, de traitement du langage naturel, etc. Elle est largement utilisée dans les projets industriels et de recherche en intelligence artificielle. [28]



Figure 15: Bibliothèques Python utilisées

5. Analyse exploratoire de données

5.1 Diagramme de Venn pour le chevauchement des catégories

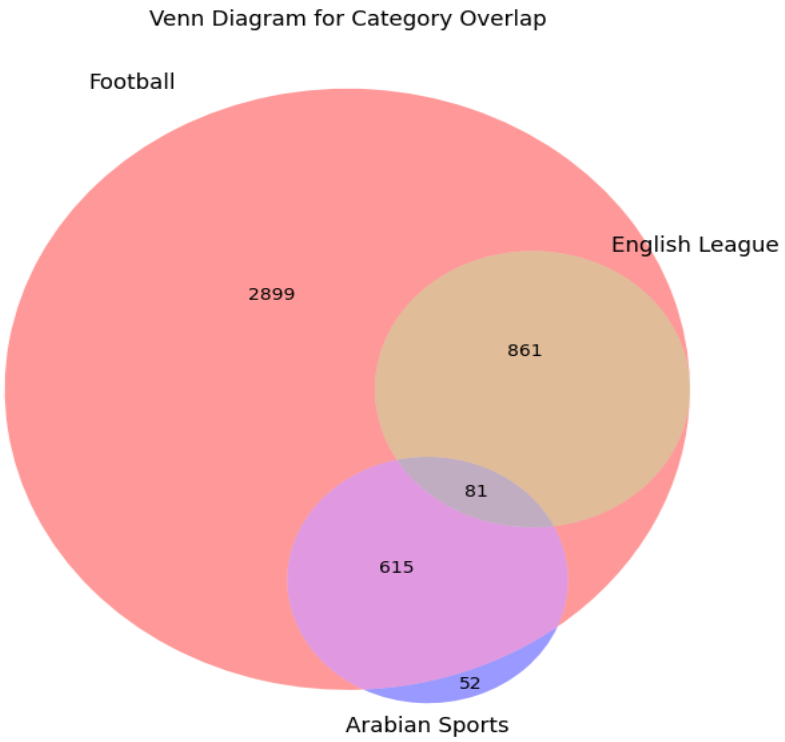

Figure 16 : Venn Diagram for Category Overlap

La figure ci-dessus présente un diagramme de Venn illustrant le chevauchement entre trois catégories thématiques : *Football*, *English League* et *Arabian Sports*. Ce type de visualisation permet de repérer visuellement les relations et les zones de cooccurrence entre ces étiquettes dans le corpus.

On constate que la catégorie *Football* est prédominante, avec un nombre élevé d'occurrences individuelles (2899 documents). Elle partage également 861 documents avec *English League*, ce qui traduit un lien thématique étroit entre ces deux catégories. De même, *Football* coexiste avec *Arabian Sports* dans 615 documents, confirmant également une certaine corrélation.

En revanche, la cooccurrence entre *English League* et *Arabian Sports* reste relativement modeste (81 documents), ce qui pourrait indiquer que ces deux thématiques sont moins souvent associées ensemble, peut-être en raison de contextes éditoriaux ou géographiques distincts.

Ce type d'analyse visuelle est précieux pour mieux cerner la structure sémantique de la base de données et peut être exploité pour ajuster des modèles de classification multi-étiquettes, en tenant compte des dépendances ou des indépendances entre les différentes classes.

5.2 Matrice de cooccurrence des étiquettes

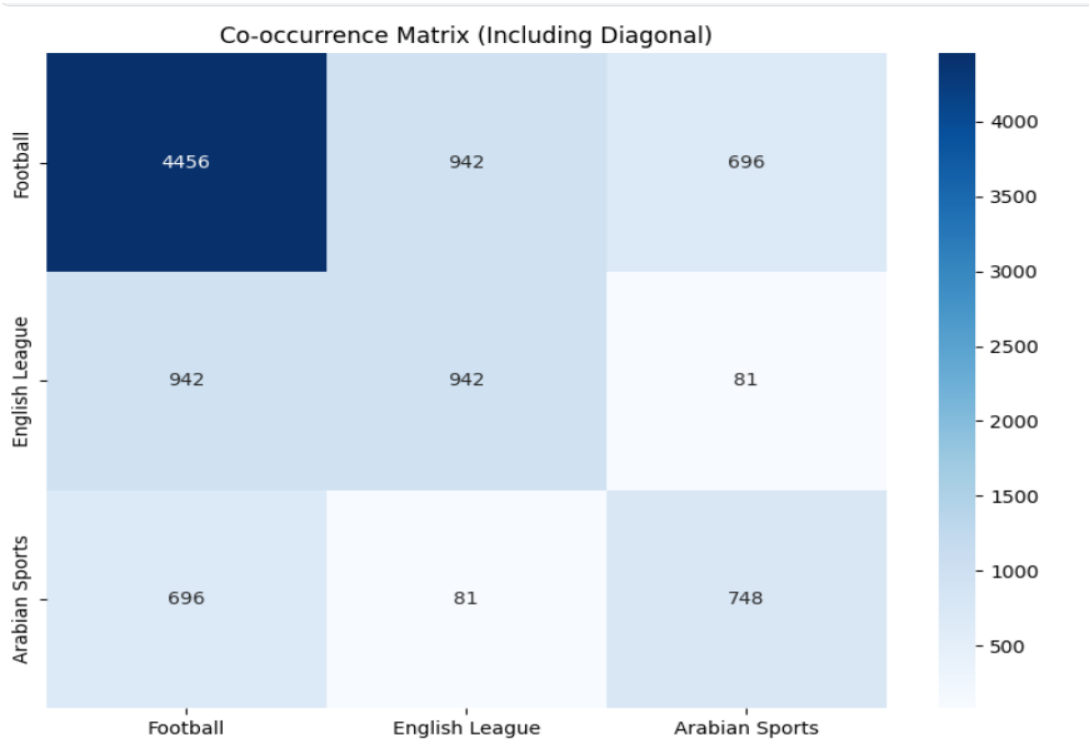

Figure 17 : Matrice de cooccurrence des étiquettes

La figure ci-dessus illustre une **matrice de cooccurrence** des étiquettes thématiques, représentant le nombre de fois où deux étiquettes apparaissent ensemble dans un même document. Cette matrice permet d'identifier les relations ou les corrélations entre différents thèmes.

On remarque par exemple que l'étiquette **"Football"** est très fréquente, avec **4456 occurrences** en diagonal (c'est-à-dire seule ou avec d'autres), ce qui montre sa forte présence dans le corpus. Elle coapparaît également **942 fois avec "English League"** et **696 fois avec "Arabian Sports"**, ce qui suggère un lien thématique fort entre ces catégories.

De plus, la cooccurrence entre **"English League"** et **"Arabian Sports"** est relativement faible (**81 occurrences**), ce qui peut indiquer une certaine spécialisation ou séparation des sujets dans les documents.

Cette analyse est utile pour mieux comprendre la structure sémantique de la base de données et peut aider à améliorer les performances d'un modèle de classification multi-étiquettes en tenant compte des dépendances entre les classes.

6. Résultats obtenus des trois modèles de classification

Cette section présente une comparaison des performances des trois modèles de classification dans le cadre de ce projet. Elle comprend également des visualisations des courbes d'apprentissage associées à chaque modèle. L'ensemble de ces résultats permet d'évaluer l'efficacité de chaque approche appliquée à la tâche de classification multi-étiquette des textes en arabes.

Modèle	Précision	Rappel	F1-Score	Exactitude	Hamming Loss
AraBERT	0.95	0.97	0.96	0.89	0.03
BiLSTM + FastText	0.98	0.72	0.83	0.64	0.12
CNN	0.96	0.92	0.94	0.85	0.04

Tableau 2 : Comparaison des performances des trois modèles

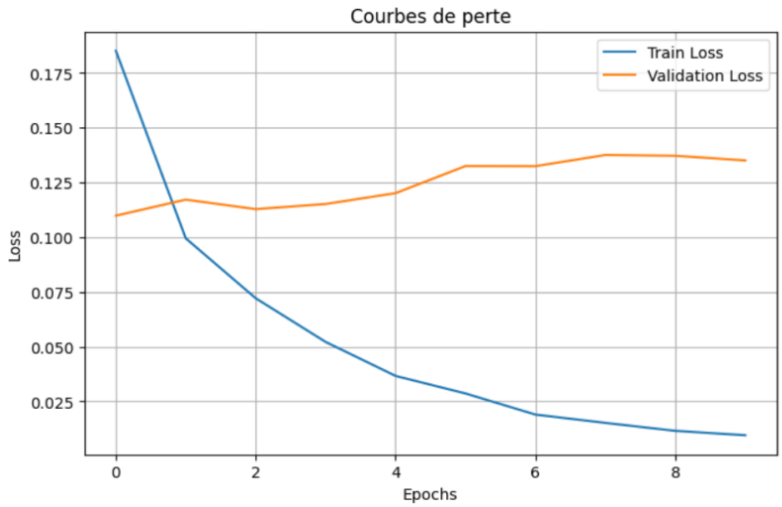

Figure 18 : perte de données selon les époques pour la méthode ARABERT

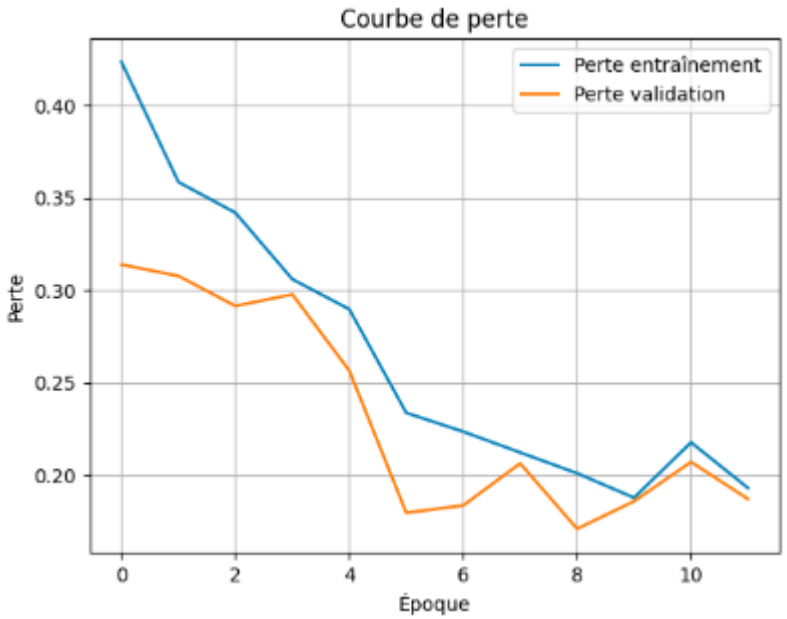

Figure 19: perte de données selon les époques pour la méthode BiLSTM

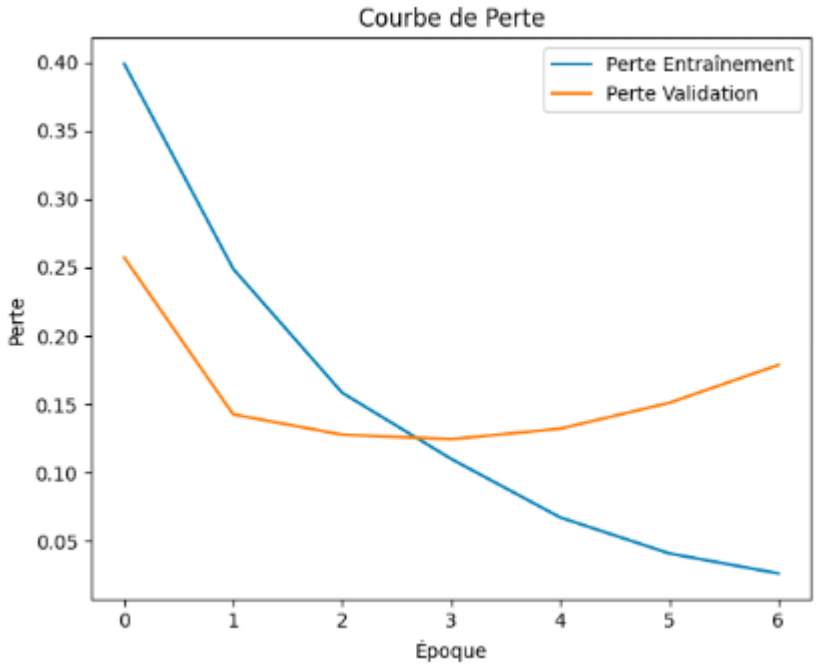

Figure 20 : perte de données selon les époques pour la méthode CNN

7. Discussion des résultats

Cette section vise à analyser et interpréter les performances obtenues par les trois modèles de classification multi-étiquette appliqués aux textes en langue arabe : AraBERT, BiLSTM avec FastText, et CNN. L'analyse se fonde sur des métriques d'évaluation standard (F1-score micro, précision, rappel, exactitude, Hamming Loss) ainsi que sur les courbes d'évolution des pertes d'entraînement et de validation.

7.1 Analyse comparative des performances

Les résultats montrent une **supériorité nette du modèle AraBERT** par rapport aux deux autres architectures. Ce modèle atteint un **F1-score micro de 0.96**, un **rappel élevé de 0.97**, ainsi qu'une **précision de 0.95**, traduisant une excellente capacité à détecter et classer correctement les différentes étiquettes. L'**exactitude** (subset accuracy) est de **0.89**, tandis que la **Hamming Loss** est remarquablement faible (0.03), indiquant une rare occurrence d'erreurs sur les étiquettes individuelles.

Le modèle **CNN** se positionne en second lieu, avec un **F1-score micro de 0.94**, une **précision de 0.96** et un **rappel de 0.92**. L'exactitude s'élève à **0.85** et la Hamming Loss reste faible à **0.04**, témoignant d'une bonne performance globale malgré l'absence de mécanisme contextuel comme celui introduit par les Transformers dans AraBERT.

Enfin, le modèle **BiLSTM avec FastText**, bien qu'ayant obtenu une **précision élevée de 0.98**, affiche un **rappel relativement bas (0.72)**, ce qui se répercute sur un **F1-score micro plus modeste (0.83)**. Son **exactitude** est de **0.64** et la **Hamming Loss** est plus élevée (0.12), ce qui suggère une difficulté à bien généraliser l'apprentissage des dépendances séquentielles dans les textes arabes.

7.2 Interprétation des courbes de perte

L'analyse des courbes de perte corrobore les métriques précédentes.

- **AraBERT** (Figure 18) présente une **diminution régulière de la perte d'entraînement**, mais une **stabilisation, voire une légère augmentation de la perte de validation** à partir de la quatrième époque. Cela suggère un début de **surapprentissage** (overfitting), bien que modéré, le modèle conservant une bonne généralisation au vu des scores obtenus.

- Pour le modèle **BiLSTM** (Figure 19), la **perte de validation diminue de manière plus significative**, avant de se stabiliser autour de l'époque 7. La convergence entre les pertes d'entraînement et de validation témoigne d'un **meilleur équilibre**, bien que la qualité des prédictions finales (notamment en rappel) reste inférieure.
- Concernant le **modèle CNN** (Figure 20), la courbe révèle également une diminution progressive des pertes, avec toutefois une **légère remontée de la perte de validation après l'époque 4**, indiquant un **début de surapprentissage**. Ce comportement est cohérent avec les scores de rappel légèrement inférieurs à ceux d'AraBERT.

7.3 Synthèse critique

En définitive, les résultats obtenus confirment la **pertinence des modèles de type Transformer** pour la classification multi-étiquette de textes arabes. Le **modèle AraBERT**, grâce à sa capacité à modéliser les dépendances contextuelles longues et à intégrer des représentations linguistiques pré-entraînées sur de larges corpus arabophones, se distingue par sa robustesse et sa précision.

Le **modèle CNN**, bien que moins performant que AraBERT, constitue une alternative valable, en particulier dans des contextes où les ressources computationnelles sont limitées. Le **modèle BiLSTM**, malgré son architecture séquentielle bidirectionnelle, peine à rivaliser, probablement en raison de la richesse morphologique de l'arabe et de la complexité des relations sémantiques que seul un encodage contextuel profond comme celui d'AraBERT semble apte à capturer.

7.4 Exploitation des modèles pour des nouveaux textes

Texte 1

تمكن المنتخب الجزائري من الفوز على نظيره التونسي في مباراة ودية أقيمت مساء أمس بالعاصمة الجزائر، حيث سجل رياض محرز هدف الفوز في الدقيقة 78.

Texte 2

توّج مانشستر سيتي بلقب الدوري الإنجليزي الممتاز بعد فوزه الساحق بنتيجة 4-0 على تشيلسي، حيث سجل إرلينغ هالاند هدفين في المباراة.

Texte 3

أعلنت رابطة الدوري الإنجليزي عن مواعيد مباريات الجولة المقبلة، فيما يستعد ليفربول لمواجهة مانشستر يونايتد في قمة مرتقبة على ملعب أنفيلد.

Les trois figures suivantes montrent les résultats de prédiction de catégories de textes en arabe en utilisant les trois meilleurs modèles de prédiction : CNN, AraBert et BiLSTM avec FastText. Voici une comparaison des performances et des différences entre ces méthodes.



Figure 21 : les résultats de prédiction obtenus en utilisant le modèle AraBert



Figure 22 : les résultats de prédiction obtenus en utilisant le modèle BiLSTM



Figure 23 : les résultats de prédiction obtenus en utilisant le modèle CNN

Conclusion

Ce chapitre a présenté les étapes essentielles de la modélisation et de l'implémentation du système de catégorisation automatique des documents. Nous y avons décrit l'architecture du système, les méthodes de classification adoptées, ainsi que les métriques d'évaluation utilisées. Les résultats obtenus, notamment avec le modèle AraBERT, CNN et BiLSTM ont permis de valider l'efficacité du système et la qualité de ses performances dans la classification de documents en langue arabe.

Conclusion générale

La classification multi-étiquette des textes en langue arabe constitue une tâche ambitieuse et complexe, en raison des particularités linguistiques de cette langue : richesse morphologique, absence de voyelles brèves dans l'écriture, diversité dialectale, et manque de ressources linguistiques bien annotées.

Dans ce mémoire, nous avons proposé une étude comparative de trois approches d'apprentissage profond appliquées à cette problématique : un modèle convolutionnel (CNN), un modèle BiLSTM avec des embeddings issus de FastText, et enfin AraBERT, un modèle préentraîné basé sur l'architecture Transformer et spécifiquement conçu pour la langue arabe. L'objectif était de déterminer quelle architecture offrait les meilleures performances en termes de classification multi-étiquette sur un sous-ensemble du corpus NADiA1, limité à trois catégories relevant du domaine sportif.

Les résultats expérimentaux ont révélé des différences notables entre les modèles. AraBERT s'est clairement imposé comme le plus performant, grâce à sa capacité à capturer finement le contexte sémantique des textes via l'apprentissage par transfert sur un vaste corpus arabe. Ce modèle a démontré un excellent équilibre entre précision et rappel, et s'est démarqué par sa capacité à reconnaître avec justesse plusieurs étiquettes pertinentes pour un même document.

Le modèle CNN a également affiché de bonnes performances globales, soulignant la pertinence de l'approche convolutionnelle pour l'extraction de motifs textuels, mais il s'est montré légèrement moins précis dans les cas ambigus.

En revanche, le modèle BiLSTM, bien que performant sur le plan de la précision, a présenté un rappel plus faible, ce qui laisse penser qu'il a manqué plusieurs étiquettes pertinentes dans les prédictions finales.

Ces résultats confirment la supériorité des modèles préentraînés comme AraBERT pour les tâches de classification de textes complexes, en particulier en arabe. Toutefois, il convient de souligner que chacun des modèles étudiés présente des avantages et des limites.

Plusieurs perspectives peuvent être envisagées pour prolonger ce travail. L'exploration de modèles hybrides combinant des architectures séquentielles et attentionnelles pourrait améliorer les performances de classification. L'extension à un plus grand nombre de catégories ainsi que l'intégration de modèles multilingues ou adaptés aux dialectes arabes renforceraient la couverture linguistique. Enfin, l'exploitation de métadonnées contextuelles et le développement d'un système de classification en temps réel ouvriraient la voie à des applications concrètes en traitement automatique des textes arabes.

Les références

1. Cambria, E., Schuller, B., Xia, Y., & Havasi, C. (2013). *New Avenues in Opinion Mining and Sentiment Analysis* https://doi.org/10.1109/MIS.2013.30

2. Habash, Nizar Y. Introduction to Arabic Natural Language Processing. Morgan & ; Claypool Publishers, 2010.

3. Sebastiani, F. (2002). Machine Learning in Automated Text Categorization. ACM Computing Surveys, 34(1), 1–47. DOI: 10.1145/505282.505283

4. © brondon styve, Tchamba Kuinze, 2024(Mémoire présenté à l'Université du Québec à Chicoutimi en vue de l'obtention du grade de Maître ès sciences (M. Sc.) en informatique (recherche))

5. https://www.shaip.com/blog/text-classification-importance-use-cases-and-process/

6. D. Marwah and J. Beel, "Term-Recency for TF-IDF, BM25 and USE Term Weighting," in *Proc. 8th Int. Workshop on Mining Scientific Publications*, Wuhan, China, 2020, pp. 36–41.

7. T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," *arXiv preprint arXiv:1301.3781*, 2013.

8. J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier Chains for Multi-label Classification," in *Proc. 13th European Conf. on Machine Learning (ECML)*, Warsaw, Poland, 2009, pp. 254–269.

9. https://wimukthimadhusanka85.medium.com/multi-label-classification-e259896182da

10. https://dspace.univguelma.dz/jspui/bitstream/123456789/12915/1/SIAFA_F5.pdf

11. S. Haykin, *Neural Networks and Learning Machines*, 3rd ed., Upper Saddle River, NJ, USA: Pearson Education, 2009.

12. C. M. Bishop, *Pattern Recognition and Machine Learning*, New York, NY, USA: Springer, 2006.

13. https://moncoachdata.com/blog/comprendre-les-reseaux-de-neurones/

14. A. Graves, "Supervised sequence labelling with recurrent neural networks," *Studies in Computational Intelligence*, vol. 385, Springer, 2012.

15. T. M. Breuel, « Benchmarking of LSTM Networks »,arXiv:1508.02774, août 2015.

16. S. J. Pan and Q. Yang, "A Survey on Transfer Learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

17. J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proc. NAACL-HLT*, Minneapolis, MN, USA, 2019, pp. 4171–4186.

18. W. Antoun, F. Baly, and H. Hajj, "AraBERT: Transformer-based Model for Arabic Language Understanding," in *Proc. 4th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT 2020)*, Language Resources and Evaluation Conference (LREC), Marseille, France, May 2020, pp. 9-15.

19. https://docs.python.org/fr/3.13/tutorial/index.html

20. https://www.datacamp.com/fr/blog/what-is-kaggle

21. McKinney, W. (2010). Data Structures for Statistical Computing in Python. In: van der Walt,S., & Millman, J. (Eds.), Proceedings of the 9th Python in Science Conference, pp. 51–56.Disponible sur : https://conference.scipy.org/proceedings/scipy2010/mckinney.html

22. Harris, C. R., Millman, K. J., van der Walt, S. J., et al. (2020). Array programming with NumPy.Nature, 585(7825), 357–362. Disponible sur : https://doi.org/10.1038/s41586-020-2649-2

23. Pedregosa, F., Varoquaux, G., Gramfort, A., et al. (2011). Scikit-learn: Machine Learning inPython. Journal of Machine Learning Research, 12, 2825–2830. Disponible sur :

http://jmlr.org/papers/v12/pedregosa11a.html

24. Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. Computing in Science &Engineering, 9(3), 90–95. Disponible sur : https://doi.org/10.1109/MCSE.2007.55

25. Paszke, A., Gross, S., Massa, F., et al. (2019). PyTorch: An Imperative Style, High-

Performance Deep Learning Library. In: Advances in Neural Information Processing

Systems, 32. Disponible sur :

https://papers.nips.cc/paper_files/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html

26. Wolf, T., Debut, L., Sanh, V., et al. (2020). Transformers: State-of-the-Art Natural Language Processing. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 38–45. Disponible sur :

https://www.aclweb.org/anthology/2020.emnlp-demos.6

27. Bird, S., Klein, E., & Loper, E. (2009). *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly Media, Inc.

28. M. Abadi et al., "TensorFlow: A system for large-scale machine learning," *OSDI*, 2016.