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

MEMOIRE

Présenté en vue de l'obtention du diplôme

Master en informatique

Spécialité : Réseaux et multimédia

THEME

La Catégorisation Multi-thématique pour le Texte Arabe

Présenté par :

GHERSSALLAH Abbes.

BESSA Khadidja.

Soutenu publiquement le : 20/06/2024.

Devant le jury composé de:

Président : SAAD SAOUD Manal.

Examinateur : FILLALI Farhat.

Encadreur : MOHDEB Djamila.

2023/2024

Dédicace

Nous dédions ce mémoire à nos chers parents, nos mères et nos pères, pour leur patience, leur amour, leur soutien et leurs encouragements, à nos frères, à nos amies et nos camarades. Sans oublier tous les professeurs que ce soit du primaire, du moyen, du secondaire ou de l’enseignement supérieur.

Remerciement

En tout premier lieu, nous remercions Dieu, tout puissant, de nous avoir donné la santé et la volonté d'entamer et de terminer ce mémoire. Ce mémoire n'aurait jamais pu voir le jour sans le soutien actif des membres de notre famille, surtout nos parents qui nous ont toujours encouragé. Enfin, nous tenons à exprimer vivement nos remerciements avec une profonde gratitude à toutes les personnes qui ont contribué de près ou de loin à sa réalisation, car un projet ne peut pas être le fruit d'une seule personne.

Résumé

Dans un contexte où l'organisation efficace des documents est cruciale en raison de la saturation informationnelle, la langue arabe présente des défis spécifiques pour la catégorisation automatique en raison de sa complexité linguistique et culturelle. L'augmentation rapide des données textuelles en arabe sur internet et sur diverses plateformes rend cette tâche indispensable mais difficile, traditionnellement réalisée manuellement et sujette aux erreurs.

Ce mémoire vise à développer une application en Python pour la catégorisation automatique multi-thématique des documents en arabe. En combinant le traitement automatique de la langue arabe (Arabic NLP) et la classification multi-label de texte, l'application cherche à surmonter les défis linguistiques et culturels propres à l'arabe, offrant ainsi un outil robuste, précis et adaptable pour améliorer la gestion et l'organisation des données textuelles en arabe.

Mots clés : Documents arabes, Catégorisation automatique, Catégorisation multi-thématique, Modélisation thématique, Algorithmes multi-label, Apprentissage automatique, Apprentissage profond.

Abstract

In a context where the effective organization of documents is crucial due to information saturation, the Arabic language presents specific challenges for automatic categorization because of its linguistic and cultural complexity. The rapid increase in Arabic textual data on the internet and various platforms makes this task indispensable yet difficult, traditionally done manually and prone to errors.

This thesis aims to develop a Python application for the automatic multi-thematic categorization of Arabic documents. By combining Arabic Natural Language Processing (Arabic NLP) and multi-label text classification, the application seeks to overcome the linguistic and cultural challenges unique to Arabic, thus offering a robust, precise, and adaptable tool to improve the management and organization of Arabic textual data.

Keywords: Arabic documents, Automatic categorization, Multi-thematic categorization, Topic modeling, Multi-label algorithms, Machine learning, Deep learning.

ملخص

في سياق حيث تكون فعالية تنظيم المستندات أمرًا حاسمًا بسبب التشبع المعلوماتي، تقدم اللغة العربية تحديات محددة للتصنيف الآلي بسبب تعقيدها اللغوي والثقافي. إن الزيادة السريعة في البيانات النصية باللغة العربية على الإنترنت وعلى منصات متنوعة تجعل هذه المهمة لا غنى عنها لكنها صعبة، إذ تُنفذ تقليديًا يدويًا ومعرضة للأخطاء.

يهدف هذا البحث إلى تطوير تطبيق باستخدام بايثون لتصنيف المستندات العربية متعدد المواضيع تلقائيًا. من خلال الجمع بين معالجة اللغة الطبيعية العربية (Arabic NLP) وتصنيف النصوص متعددة التسميات، يسعى التطبيق إلى التغلب على التحديات اللغوية والثقافية الفريدة للعربية، مما يوفر أداة قوية ودقيقة وقابلة للتكيف لتحسين إدارة وتنظيم البيانات النصية العربية.

الكلمات المفتاحية: المستندات العربية، التصنيف الآلي، التصنيف متعدد المواضيع، نمذجة المواضيع، خوارزميات متعددة التسميات، التعلم الآلي، التعلم العميق.

.

Table des matières

Liste des abréviations ..... **ix**

Liste des figures ..... **x**

Liste des tableaux ..... **xii**

Liste des algorithmes ..... **xiii**

Introduction Générale **1**

1. Contexte **1**

2. Problématique **1**

3. Objectifs **1**

4. Structure du rapport **2**

Chapitre 01 : Catégorisation des documents arabes **3**

1.1 Introduction 3
1.2 Définition de la catégorisation des documents 3
1.3 Application de la catégorisation des documents 4
1.4 Processus de catégorisation des documents 5
1.5 Approches de catégorisation de documents pour la langue arabe 6
1.5.1 Approches classiques 6
1.5.2 Approches modernes basées sur le traitement automatique du langage naturel 7
1.6 Défis majeures de la catégorisation des documents arabe 8
1.7 Conclusion 9

Chapitre 02 : La classification multi-label de texte **10**

2.1 Introduction 10
2.2 Définition 10
2.3 Processus de classification multi-label du texte 10
2.3.1 Collecte et préparation des données 11
2.3.2 Représentation du texte 11

2.3.3 Transformation ou adaptation du problème de classification multi-label 11
2.3.4 Choix du modèle 11
2.3.5 Division des données 12
2.3.6 Entraînement du modèle 12
2.3.7 Optimisation des hyperparamètres 12
2.3.8 Évaluation du modèle 12
2.3.9 Interprétation des résultats 12
2.4 Les approches de transformation du problème de classification multi-label 13
2.4.1 Pertinence binaire (Binary Relevance BR) 13
2.4.2 Ensemble de puissances d'étiquettes (Label Powerset LP) 13
2.4.3 Chaîne de classificateurs (Classifier chains CC) 14
2.5 Les méthodes de la classification multi-label 15
2.5.1 Les méthodes de base 15
2.5.2 Méthodes ensemblistes 16
2.5.3 Méthodes Deep Learning 17
2.6 Les défis de la classification multi-label 18
2.7 Conclusion 19

Chapitre 03 : Architecture et Modélisation 20

3.1 Introduction 20
3.2 Description de projet 20
3.3 Fonctionnalités du système de catégorisation des documents 22
3.3.1 Chargement de données 22
3.3.2 Nettoyage et prétraitement de données 22
3.3.3 Visualisation 23
3.3.4 Vectorisation du texte 23
3.3.5 Catégorisation des documents 24
3.3.6 L'évaluation 27
3.4 Conclusion 28

Chapitre 04 : Implémentation et Résultats 29

4.1 Introduction 29
4.2 Environnement de travail et outils d'implémentation 29
4.2.1 Matériel 29
4.2.2 Environnement de programmation 29

4.3 Description d'application 31
4.4 Cas d'étude : Catégorisation des documents arabes 35
4.4.1 Description du jeu de données 35
4.4.2 Analyse exploratoire de données 37
4.4.3 Prétraitement 40
4.4.4 Vectorisation 41
4.4.5 Catégorisation automatique des documents arabes 41
4.4.6 Discussion des résultats 44
4.5 Conclusion 47
Conclusion générale 48
Les références 49

Liste des abréviations

NLTK	Natural Language Toolkit (Traitement Automatique du Langage Naturel)
BDD	Base De Données
K-NN	K-Nearest Neighbors (k plus proches voisins)
BR	Binary Relevance (Pertinence binaire)
CC	Classifier Chains (Chaînes de classificateurs)
LP	Label Powerset (Ensemble d'étiquettes)
CNN	Convolutional Neural Network (Réseau de neurones convolutif)
RNN	Recurrent Neural Network (Réseau de neurones récurrents)
NLP	Natural Language Processing

Liste des figures

Figure 1: Un exemple de transformation d'un problème de classification multi-étiquettes à un problème de classification binaire en utilisant l'approche de pertinence binaire 13

Figure 2 : Approches d'étalonnage d'étiquettes 14

Figure 3 : Approches chaine de classifieurs. 14

Figure 4: Architecture du système proposé..... 21

Figure 5 : Structure d'une base de données multi-étiquettée. 22

Figure 6 : Bibliothèques Python utilisées 30

Figure 7 : Interface principale de l'application 32

Figure 8 : Interface de chargement et visualisation des données 33

Figure 9 : Interface de prétraitement de données..... 33

Figure 10 : Interface de catégorisation des documents à l'aide des classificateurs de base. . 34

Figure 11 : Interface de catégorisation des documents à l'aide des modèles d'apprentissage profond..... 34

Figure 12 : Interface de détection automatique des thématiques des documents à l'aide des modèles déjà entraînés 35

Figure 13 : Distribution des catégories. 37

Figure 14 : Le nuage de mots du jeu de données 38

Figure 15 : Les 1-grammes les plus fréquents dans le jeu de données 39

Figure 16 : Les bi-grammes les plus fréquents dans le jeu de données 40

Figure 17 : Texte avant prétraitement 40

Figure 18 : Texte après prétraitement 40

Figure 19 : Texte avant vectorisation 41

Figure 20 : Texte après vectorisation..... 41

Figure 21 : Exactitude et perte de données selon les époques pour la méthode CNN..... 43

Figure 22: Exactitude et perte de données selon les époques pour la méthode RNN..... 43

Liste des tableaux

Tableau 1: Table de matériels 29

Tableau 2: Caractéristiques de l'ensemble de données 37

Tableau 3 : Performance des classificateurs de base avec l'approche BR 42

Tableau 4 : Performance des classificateurs de base avec l'approche LP 42

Tableau 5 : Performance des classificateurs de base avec l'approche CC 42

Tableau 6 : Performance des classificateurs d'apprentissage profond 42

Liste des algorithmes

Algorithme 1: Algorithme BR. 25

Algorithme 2 : Algorithme LP 25

Algorithme 3: Algorithme CC. 26

Introduction Générale

1. Contexte

Dans un environnement saturé d'informations, classer et organiser les documents de manière efficace est essentiel. Cette tâche est particulièrement complexe pour la langue arabe, en raison de ses spécificités linguistiques et culturelles. D’un autre côté, l'augmentation rapide des données textuelles en arabe, que ce soit sur les plateformes en ligne, dans les documents gouvernementaux ou les publications académiques, rend la catégorisation précise et efficace indispensable. Réalisée traditionnellement à la main, cette tâche est non seulement fastidieuse mais aussi sujette aux erreurs humaines. Les avancées en apprentissage automatique et en intelligence artificielle offrent désormais une solution automatisée et fiable à ce défi.

2. Problématique

La catégorisation automatique des documents arabes pose toutefois des défis uniques. Parmi eux, on compte la complexité intrinsèque de la langue arabe, avec ses racines trilitères, ses variations dialectales et ses écritures cursives. De plus, le manque de ressources linguistiques annotées et de modèles pré-entraînés spécifiques à l'arabe complique le développement de systèmes efficaces de catégorisation automatique.

3. Objectifs

Ce mémoire vise à concevoir et à développer une application de catégorisation automatique des documents arabes qui surmonte ces défis spécifiques. En combinant le traitement automatique de la langue arabe (Arabic NLP) et la classification multi-label de texte (Multi-label Text Classification), notre objectif est de créer, sous Python, une application robuste, précise et adaptable capable de classer et de catégoriser efficacement une variété de documents arabes.

En fournissant un outil précieux pour la gestion et l'organisation de grandes quantités de données textuelles en arabe, notre travail vise à faciliter l'accès à l'information et à améliorer l'efficacité des processus de traitement des documents dans un contexte arabophone.

4. Structure du rapport

La suite de ce mémoire est organisée en cinq chapitres :

- Le premier chapitre aborde la catégorisation multi-thématique, en expliquant ses concepts clés.
- Le deuxième chapitre explore la classification multi-label du texte, détaillant les processus, approches, méthodes et défis associés.
- Le troisième chapitre décrit la méthodologie utilisée pour concevoir et structurer la partie pratique du projet.
- Le quatrième et dernier chapitre décrit l'environnement matériel et logiciel utilisé pour implémenter notre application tout au long avec un cas d’étude, et présente une discussion des résultats obtenus.

Chapitre 01 : Catégorisation des documents arabes

1.1 Introduction

La catégorisation dans les corpus arabes constitue un domaine de recherche en constante évolution. En tant que composante essentielle du traitement automatique du langage naturel (TALN), elle vise à extraire des informations significatives à partir de grandes quantités de données textuelles, facilitant ainsi la compréhension des tendances, des préoccupations et des sujets clés dans des contextes divers.

Dans ce chapitre, nous nous aborderons les concepts de base de la catégorisation des documents, ses applications, ses processus, ses approches classics et modernes et ses défis majeurs dans le contexte de la langue arabe.

1.2 Définition de la catégorisation des documents

La catégorisation des documents, également connue sous le nom de modélisation thématique (Topic Modeling) ou encore catégorisation thématique, est un processus d'organisation et de classification automatique ou semi-automatique de documents textuels en groupes ou catégories basés sur leurs contenus sémantiques. Formellement, on peut la définir comme suit :

Soit $D = \{d_1, d_2, d_3, \dots., d_n\}$ un ensemble de documents textuels où d_i représente un document, et soit $C = \{c_1, c_2, c_3, \dots, c_m\}$ l'ensemble de catégories prédéfinies où c_j représente une catégorie. La catégorisation des documents consiste à attribuer à chaque document $d_i$une ou plusieurs catégories c_j de manière automatique ou semi-automatique en utilisant des techniques d'apprentissage automatique (ML) ou des méthodes de traitement automatique du langage naturel (NLP) [1] [2].

1.3 Application de la catégorisation des documents

La catégorisation des documents revêt une importance cruciale dans de nombreux domaines pour plusieurs raisons [2] [3] :

- **Gestion de l'information**

Elle permet d'organiser efficacement de vastes quantités de documents textuels, facilitant ainsi la recherche, la récupération et la navigation dans l'information [2] [3].

- **Veille stratégique**

En identifiant et en catégorisant automatiquement les tendances, les sujets et les événements dans les documents, elle aide les organisations à rester informées des développements pertinents dans leur domaine d'intérêt [2] [3].

- **Personnalisation du contenu**

Elle permet de recommander du contenu pertinent aux utilisateurs en fonction de leurs intérêts et préférences, améliorant ainsi l'expérience utilisateur [2] [3].

- **Analyse de sentiment**

En identifiant les thèmes et les sentiments exprimés dans les documents, elle aide à comprendre les opinions, les attitudes et les perceptions des utilisateurs à l'égard de certains sujets [2] [3].

- **Prise de décision**

Elle fournit des informations organisées et structurées qui peuvent être utilisées pour prendre des décisions éclairées dans divers domaines tels que la politique, les affaires et la recherche [2] [3].

- **Analyse de la recherche académique**

Catégorisation automatique des articles de recherche en fonction de leurs domaines d'étude et de leurs sujets [2] [3].

1.4 Processus de catégorisation des documents

Le processus de catégorisation de documents implique généralement les étapes suivantes [4] [5] :

- **Prétraitement des données**

Cela comprend la suppression des mots vides, la lemmatisation, la racinisation et d'autres techniques de normalisation de texte [4] [5].

- **Représentation des documents**

Les documents sont représentés sous forme de vecteurs numériques à l'aide de techniques telles que la représentation de sacs de mots (Bag-of-Words) ou les embeddings de mots [4] [5].

- **Choix de l'algorithme de classification**

Des algorithmes d'apprentissage automatique tels que les machines à vecteurs de support (SVM), les arbres de décision, et les réseaux neuronaux sont souvent utilisés pour la classification des documents [4] [5].

- **Entraînement du modèle**

Le modèle est entraîné sur un ensemble de données annotées, où chaque document est associé à une ou plusieurs catégories [4] [5].

- **Évaluation du modèle**

Le modèle est évalué en utilisant des mesures telles que la précision, le rappel et la F-mesure pour évaluer sa performance [4] [5].

- **Application du modèle**

Une fois entraîné et évalué, le modèle peut être utilisé pour classer automatiquement de nouveaux documents en fonction des catégories prédéfinies [4] [5].

1.5 Approches de catégorisation de documents pour la langue arabe

1.5.1 Approches classiques

La catégorisation des documents dans les corpus arabes a historiquement reposé sur des approches traditionnelles, notamment [6] :

- **Analyse manuelle des documents arabes**

L’analyse manuelle implique l’intervention humaine pour identifier, catégoriser et analyser les thèmes. Elle offre une compréhension approfondie des nuances culturelles et contextuelles, mais peut être sujette à la subjectivité et être chronophage [6].

- **Techniques de catégorisation basées sur des lexiques prédéfinis**

Cette approche utilise des lexiques prédéfinis pour catégoriser les thèmes, mais elle peut manquer de flexibilité et de capacité d'adaptation à l'évolution rapide du langage et des thèmes [6].

- **Approche taxonomique**

Création d’une taxonomie hiérarchique des thèmes potentiels dans un corpus arabe [6].

- **Analyse de fréquence de mots clés**

Identification des mots clés les plus fréquemment utilisés pour déterminer les thèmes prévalents [6].

- **Analyse sémantique**

Découverte des relations sémantiques cachées entre les mots dans un corpus, mais cette approche peut nécessiter une modélisation complexe.

Ces approches traditionnelles ont des limitations en termes de subjectivité, de rigidité, de complexité linguistique et de scalabilité, ce qui a conduit à l'exploration de méthodes plus modernes, notamment celles basées sur le traitement automatique du langage naturel (TALN)[6].

1.5.2 Approches modernes basées sur le traitement automatique du langage naturel

L'évolution des technologies a introduit des approches modernes basées sur le Traitement Automatique du Langage Naturel (TALN) dans la catégorisation thématique des corpus arabes. Ces approches exploitent des méthodes plus avancées pour analyser et comprendre les textes de manière automatisée. Voici quelques-unes de ces approches [6] :

- **Utilisation de Modèles Pré-entrainés pour la Langue Arabe**

Les modèles pré-entrainés, tels que les embeddings de mots basés sur des modèles de langage comme BERT (Bidirectional Encoder Representationsfrom Transformers), sont adaptés à la langue arabe. Ces modèles captent la sémantique et la contexture des mots, permettant une représentation plus riche des thèmes [6].

- **Applications de l'Apprentissage Automatique**

L'intégration d'algorithmes d'apprentissage automatique, tels que les classificateurs, permet d'automatiser la catégorisation thématique en apprenant à partir de données annotées. Les modèles peuvent être entrainés sur des corpus arabes pour prédire les catégories dans de nouveaux documents [6].

1.6 Défis majeures de la catégorisation des documents arabe

La catégorisation automatique des documents arabes présente plusieurs défis majeurs, en particulier lorsqu'il s'agit de la catégorisation multi-thématique, où un document peut appartenir à plusieurs catégories simultanément. Voici quelques-uns de ces défis [7] :

- **Complexité linguistique**

La langue arabe présente une complexité linguistique unique, notamment en raison de sa structure morphologique complexe, de ses variations dialectales et de son écriture cursive. Ces caractéristiques rendent la segmentation des mots, l'extraction des racines et la normalisation du texte plus difficiles par rapport aux langues à structure plus simple [7].

- **Manque de ressources annotées**

Le manque de grandes quantités de données annotées en langue arabe constitue un défi majeur pour l'entraînement des modèles de catégorisation automatique. Les données annotées sont nécessaires pour l'apprentissage supervisé, mais leur disponibilité est limitée par rapport à d'autres langues comme l'anglais [7].

- **Variabilité des thèmes et des sujets**

Les documents arabes peuvent couvrir une grande variété de sujets et de thèmes, allant de la politique et de l'économie à la culture et à la religion. Cette diversité de sujets rend la catégorisation multi-thématique encore plus complexe, car un document peut contenir des informations pertinentes pour plusieurs catégories simultanément [7].

- **Défis de la catégorisation multi-thématique**

Contrairement à la catégorisation mono-thématique où un document est attribué à une seule catégorie, la catégorisation multi-thématique implique de prédire plusieurs catégories pour un même document. Cela nécessite des modèles et des techniques plus avancés pour capturer les relations complexes entre les différents thèmes présents dans un document [7].

- **Adaptation des modèles existants**

De nombreux modèles et techniques de catégorisation automatique ont été développés en se concentrant principalement sur des langues telles que l'anglais. Adapter ces modèles à la langue arabe peut nécessiter des ajustements significatifs pour tenir compte des spécificités linguistiques et culturelles de l'arabe [7].

- **Taille limitée des corpus de données**

En raison du manque de ressources annotées et de la complexité de la collecte de données en langue arabe, les corpus de données disponibles pour l'entraînement des modèles peuvent être relativement petits par rapport à d'autres langues. Cela peut entraîner des problèmes de généralisation et de performance des modèles sur des données réelles [7].

1.7 Conclusion

La catégorisation des documents pour les corpus arabes constitue un domaine dynamique, confronté aux spécificités linguistiques de l'arabe. Les approches traditionnelles présentent des limites, encourageant l'émergence d'approches modernes basées sur le TALN. Pour l'évolution il faut promet une compréhension approfondie des contenus arabes, intégrant les avancées technologiques et respectant les nuances linguistiques et culturelles

2 Chapitre 02 : La classification multi-label de texte

2.1 Introduction

Ce chapitre explore la classification multi-label de texte, une technique permettant d'assigner plusieurs étiquettes à un même document. Nous aborderons les principes fondamentaux, les méthodes couramment utilisées, et les défis spécifiques liés à cette approche.

2.2 Définition

La classification multi-label 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 une seule 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 [8].

2.3 Processus de classification multi-label du texte

Le processus de classification multi-label implique plusieurs étapes, allant de la préparation des données à l'évaluation du modèle [8] [9].

2.3.1 Collecte et préparation des données

Cela consiste à collecter un ensemble de données étiqueté, où chaque exemple est associé à une ou plusieurs classes puis nettoyer et prétraiter les données en effectuant des opérations telles que la tokenisation, la suppression des stop words, la lemmatisation, etc.

2.3.2 Représentation du texte

Cela consiste à convertir le texte en une représentation numérique. Les techniques courantes incluent la représentation en sac de mots (BoW), les embeddings de mots (comme Word2Vec ou GloVe), ou l'utilisation de modèles de langage pré-entraînés.

2.3.3 Transformation ou adaptation du problème de classification multi-label

La transformation ou adaptation du problème de classification multi-label est une étape cruciale pour rendre ce type de classification plus gérable et efficace. En raison de la complexité inhérente à l'attribution de plusieurs étiquettes à une même donnée, il est nécessaire de reformuler le problème de manière à le rendre compatible avec des algorithmes de classification existants.

Pour transformer ou adapter le problème de classification multi-label, trois approches principales sont couramment utilisées : pertinence binaire, chaines de classificateurs et ensemble de puissances des étiquettes.

2.3.4 Choix du modèle

Cela consiste à sélectionner un modèle adapté à la tâche de de classification multi-label. Cela peut inclure des modèles linéaires tels que la régression logistique, des modèles d'arbres de décision, des machines à vecteurs de support (SVM) ou des modèles d'apprentissage profond tels que les réseaux neuronaux.

2.3.5 Division des données

Cela consiste à diviser l'ensemble de données en ensembles d'entraînement, de validation et de test. L'ensemble d'entraînement est utilisé pour former le modèle, l'ensemble de validation est utilisé pour ajuster les hyperparamètres, et l'ensemble de test évalue la performance finale du modèle.

2.3.6 Entraînement du modèle

Cela consiste à alimenter le modèle avec l'ensemble d'entraînement et ajuster ses poids afin qu'il puisse apprendre les motifs associés aux différentes classes.

2.3.7 Optimisation des hyperparamètres

Cela consiste à ajuster les hyperparamètres du modèle en utilisant l'ensemble de validation pour améliorer ses performances.

2.3.8 Évaluation du modèle

Cela consiste à évaluer la performance du modèle sur l'ensemble de test en utilisant des métriques appropriées telles que la précision, le rappel, la F-mesure, etc.

2.3.9 Interprétation des résultats

Cela consiste à analyser les prédictions du modèle pour comprendre ses forces et ses faiblesses. Cela peut inclure l'inspection des erreurs de classification et la compréhension des cas où le modèle a du mal.

2.4 Les approches de transformation du problème de classification multi-label

2.4.1 Pertinence binaire (Binary Relevance BR)

Cette technique traite chaque étiquette indépendamment, et les multi-étiquettes sont ensuite séparées en classification à classe unique.

Un ensemble de classificateurs binaires simplement étiquetés est formé séparément sur l'ensemble de données d'origine pour prédire l'appartenance à chaque classe [10][11].

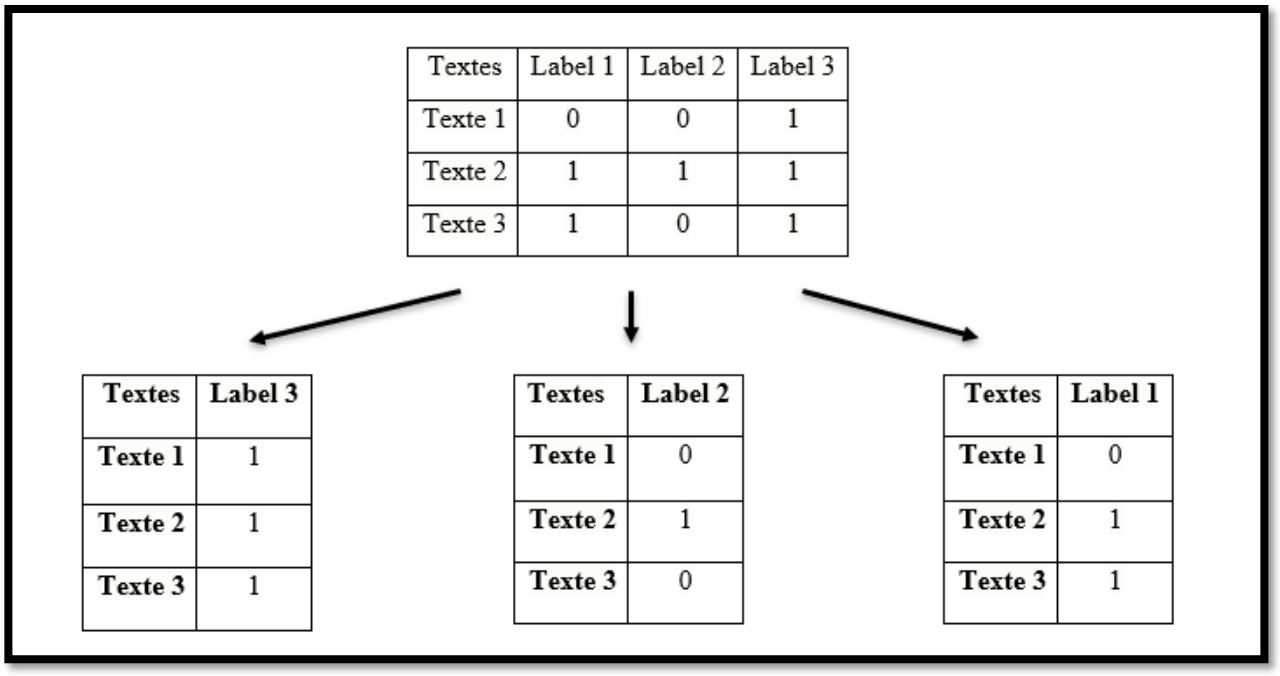

Figure 1: Un exemple de transformation d'un problème de classification multi-étiquettes à un problème de classification binaire en utilisant l'approche de pertinence binaire

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

Elle convertit un problème d'apprentissage multi-label en un seul problème d'apprentissage multi-classe à étiquette unique. Toute combinaison d'étiquettes présentes dans l'ensemble d'apprentissage sont transformés en classes, puis on forme un classificateur multi-classe .

L’objectif du ensemble de puissances d'étiquettes est de trouver une combinaison d’étiquettes uniques et de leur attribuer des valeurs différentes [10][11].

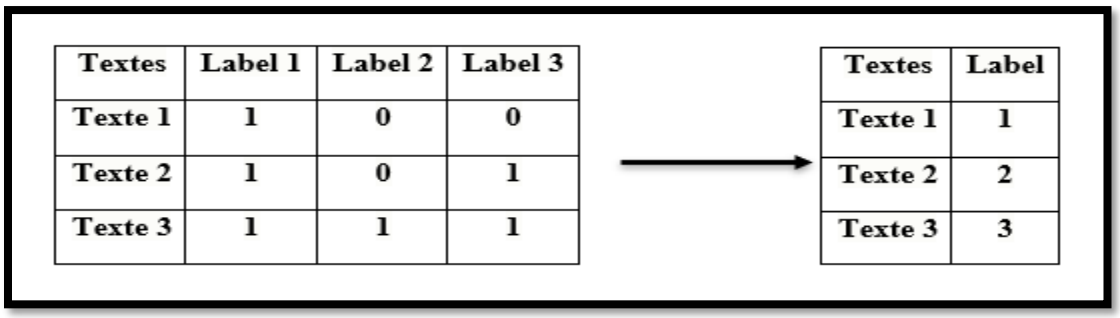

Figure 2 : Approches d'étalonnage d'étiquettes .

2.4.3 Chaîne de classificateurs (Classifier chains CC)

Le modèle de chaîne de classificateurs apprend des classificateurs comme dans la méthode de pertinence binaire. Cependant, tous les classificateurs sont liés dans une chaîne.

Il s'agit d'un processus séquentiel dans lequel la sortie d'un classificateur est utilisée comme entrée du classificateur suivant dans la chaîne [10][11].

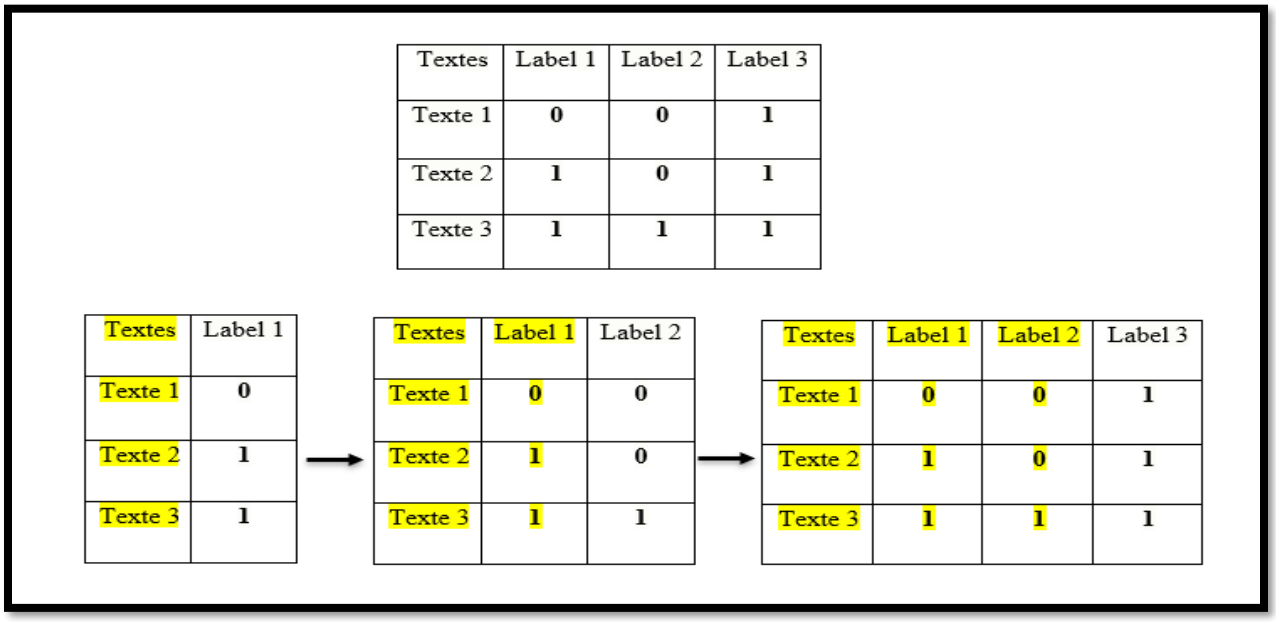

Figure 3 : Approches chaine de classifieurs.

2.5 Les méthodes de la classification multi-label

Une fois le problème de classification multi-label est adapté à un problème de classification classique, de nombreuses méthodes d'apprentissage automatique peuvent être exploitées et chacune présente ses propres caractéristiques et avantages. Voici quelques-unes des méthodes couramment utilisées :

2.5.1 Les méthodes de base

- **Régression Logistique**

La régression logistique, malgré son nom, est un modèle linéaire de classification plutôt que de régression. La régression logistique est également connue dans la littérature sous le nom de régression logistique, classification d'entropie maximale (MaxEnt) ou classificateur log-linéaire. Dans ce modèle, les probabilités décrivant les résultats possibles d'un seul essai sont modélisées à l'aide d'une fonction logistique [12].

- **Machines à Vecteurs de Support (SVM)**

La fonction de décision des SVM dépend d'un sous-ensemble des données d'apprentissage, appelé vecteurs de support, une fois ajusté, le modèle peut être utilisé pour prédire de nouvelles valeurs.

Le classificateur de Vecteur de Support Linéaire implémente une stratégie multiclasses « un contre le reste », entraînant ainsi des modèles n_class, s'il n'y a que deux classes, un seul modèle est formé [13].

- **Naïve de Bayes**

La classification naïve bayésienne est un type de classification bayésienne probabiliste simple basée sur le théorème de Bayes avec une forte indépendance (dite naïve) des hypothèses [13].

- **Méthodes de Plus Proches Voisins (K-NN)**

Un algorithme k-plus proche voisin, souvent abrégé k-NN, est une approche de la classification des données qui estime la probabilité qu'un point de données soit membre d'un groupe ou de l'autre selon le groupe dans lequel se trouvent les points de données les plus proches.

2.5.2 Méthodes ensemblistes

- **Arbre de décision (Decision Tree)**

L'arbre de décision est un algorithme qui se base sur un modèle de graphe (les arbres) pour définir la décision finale. Chaque nœud comporte une condition, et les branchements sont en fonction de cette condition (Vrai ou Faux). Plus on descend dans l'arbre, plus on cumule les conditions [8].

- **Forêts aléatoire (Random Forest)**

Une forêt aléatoire est un méta-estimateur qui ajuste un certain nombre de classificateurs d'arbres de décision sur divers sous-échantillons de l'ensemble de données et utilise la moyenne pour améliorer la précision prédictive et contrôler le surajustement. Les forêts aléatoires sont une combinaison de plusieurs arbres de décision. Chaque arbre utilise un ensemble de valeurs aléatoires qui sont choisies de manière indépendante et avec la même distribution pour tous les arbres de la forêt. Les arbres de la forêt utilisent la meilleure stratégie de division et, en utilisant une sélection aléatoire de caractéristiques pour diviser chaque nœud, les forêts aléatoires obtiennent des taux d'erreur qui sont souvent meilleurs que ceux d'Adaboost [14][15].

- **Boosting (AdaBoost)**

Le boosting est une méthode qui combine un ensemble d'apprenants faibles en un apprenant fort, afin de réduire les erreurs d'apprentissage. Dans le boosting, un échantillon aléatoire de données est sélectionné, doté d'un modèle, puis entraîné séquentiellement, c'est-à-dire que chaque modèle tente de compenser les faiblesses de son prédécesseur [16].

2.5.3 Méthodes Deep Learning

- **Réseaux de Neurones Artificiels**

Un réseau de neurone Artificiel peut être considéré comme une boite noire, qui reçoit des signaux d'entrée et produit des signaux de sortie c'est un modèle mathématique composé d'un grand nombre d'éléments de calculs organisée sous forme de couches interconnectées [17][18].

- **Réseaux de Neurones Transformer**

Ces architectures, introduites pour la première fois dans le domaine de la traduction automatique, ont également été efficacement utilisées pour la multi-label. Ils exploitent des mécanismes d'attention pour capturer les relations entre différentes parties des séquences [17].

- **Les Réseaux Neuronaux Récurrents (Recurrent Neural Networks RNNs)**

Les réseaux de neurones récurrents (RNN) sont une classe de réseaux de neurones utilisés pour modéliser les données de séquence. Dérivés des réseaux de neurones à propagation avant, les RNN présentent un comportement similaire au fonctionnement du cerveau humain. En termes simples : les réseaux de neurones récurrents produisent des résultats prédictifs dans des données séquentielles que d'autres algorithmes ne peuvent pas le faire [19].

- **Les Réseaux de Neurones Convolutifs (Convolutional Neural Networks CNN)**

CNN, également connu sous le nom de ConvNets, est un algorithme d'apprentissage en profondeur populaire et trouve de nombreux cas d'utilisation dans le domaine de la reconnaissance d'images et de la détection d'objets. Convolution signifie une opération mathématique sur deux fonctions pour donner une fonction résultante qui exprime le changement en raison de l'opération effectuée [19].

- **Les Réseaux de Mémoire à Long Terme et Court Terme (Long Short-Term Memory Networks LSTMs)**

Les LSTM sont un type spécial de RNN et sont très capables d'apprendre les dépendances à long terme. Le modèle LSTM est constitué de différents blocs de mémoire appelés cellules (les blocs rectangulaires). L'état de la cellule et l'état caché sont transférés à la cellule suivante. Comme son nom l'indique, les blocs de mémoire se souviennent des choses. Les modifications de ces blocs de mémoire se font par des mécanismes appelés portes [8].

2.6 Les défis de la classification multi-label

- L'obstacle de trouver des modèles efficaces pour résoudre des problèmes de classification multi-label, car ils ont tendance à être plus complexes que les modèles de classification mono-label [20].
- La difficulté d'évaluer les performances des modèles de classification multi-label, car il existe plusieurs façons de mesurer la précision, le rappel et le score F1. Cette diversité d'évaluations peut rendre l'interprétation des résultats plus complexe et pose un défi pour la comparaison des performances entre différents modèles.
- Il est plus difficile de généraliser les résultats des modèles de classification multi-label à de nouvelles données, car ils ont tendance à être plus sensibles aux variations des données d'entraînement.
- La complexité du modèle peut augmenter considérablement avec un nombre élevé d'étiquettes.
- Les algorithmes de classification multi-label peuvent être plus difficiles à évaluer et à optimiser.
- Les algorithmes de classification multi-label peuvent être plus coûteux en termes de temps de calcul et de mémoire.

- Problème du déséquilibre des classes, Cela peut entraîner un biais dans les modèles de classification, où les étiquettes minoritaires sont souvent moins bien prédites.
- Problème de la corrélation entre les étiquettes. Les étiquettes peuvent être liées les unes aux autres et présenter une certaine dépendance.

2.7 Conclusion

La classification multi-label est une tâche complexe mais cruciale dans le domaine de l'apprentissage automatique et du traitement automatique du langage naturel. Face à la diversité des défis tels que le déséquilibre de classe, l'ambiguïté entre les classes, et d'autres, les chercheurs et les praticiens doivent adopter des approches stratégiques pour obtenir des modèles performants et robustes.

3 Chapitre 03 : Architecture et Modélisation

3.1 Introduction

Dans cette section, nous introduisons notre projet visant à développer un système de catégorisation automatique des documents en langue arabe. Notre objectif est de concevoir une application robuste et efficace qui permettra d'analyser et de classifier un large corpus de documents arabes selon différents thèmes. Pour ce faire, nous détaillerons la méthodologie utilisée, en mettant en avant les différentes étapes de développement de notre système, ainsi que les techniques et approches algorithmiques adoptées pour la catégorisation multi-thématique des documents.

3.2 Description de projet

L’objectif de ce projet est de développer un système sophistiqué capable de comprendre et d’attribuer des étiquettes à des textes en arabe en fonction de plusieurs catégories simultanément. Pour ce faire, une application Python est conçue pour faciliter la catégorisation des documents arabes en utilisant une approche de classification multi-étiquettes et en intégrant un ensemble complet de fonctionnalités, allant du chargement et du prétraitement des données aux techniques avancées d'apprentissage automatique et profond pour la classification des documents. L'application comprend également des outils robustes pour la visualisation des données et l'évaluation des classificateurs afin de garantir une analyse approfondie et des insights pertinents.

Les fonctionnalités proposées par notre système de catégorisation automatique des documents arabe est illustrées dans la figure 4 ci-dessous.

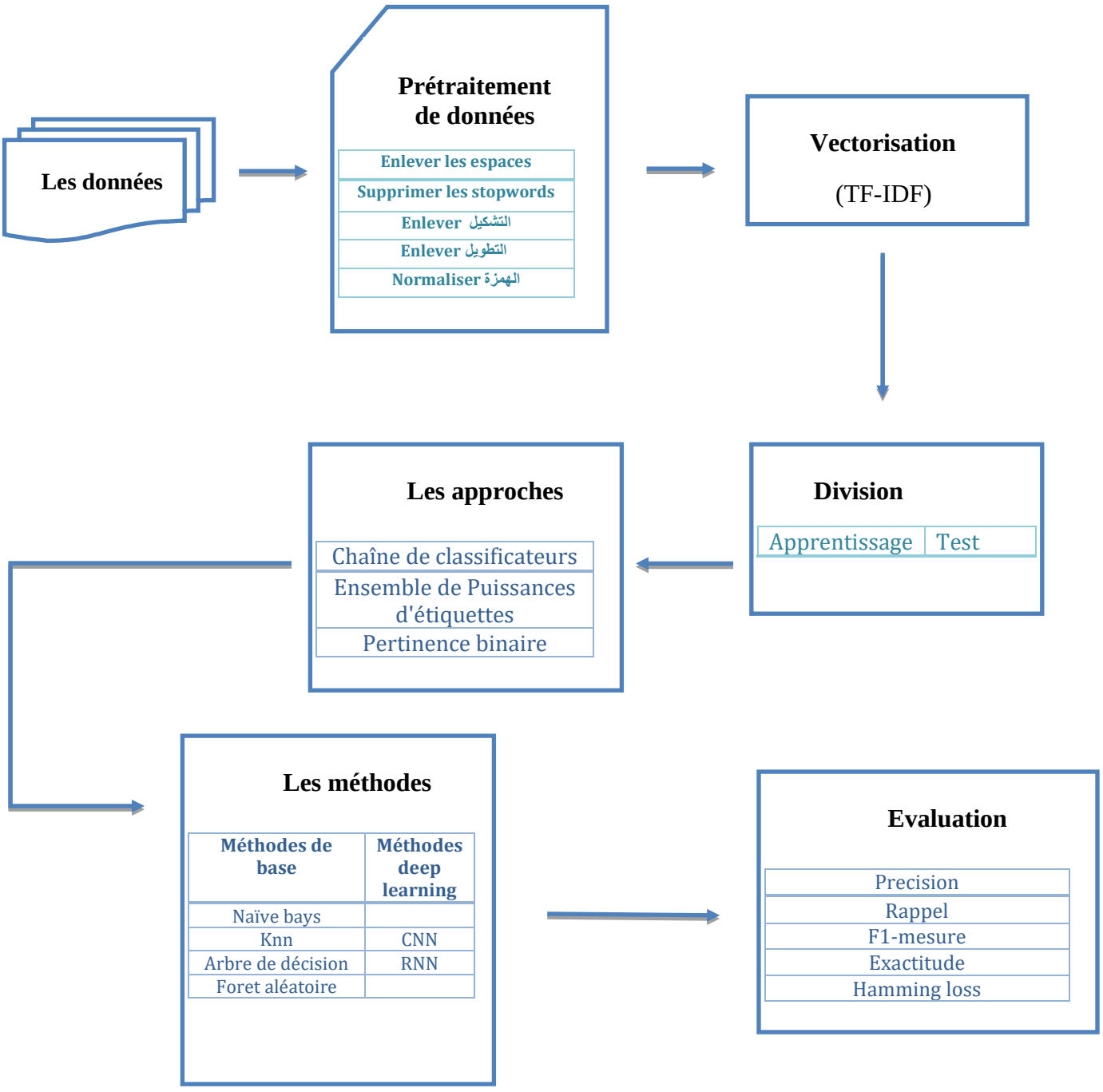

Figure 4: Architecture du système proposé.

3.3 Fonctionnalités du système de catégorisation des documents

3.3.1 Chargement de données

Le système développé permet d'importer des données depuis divers formats (CSV, Excel), principalement une base de données de documents arabes multi-étiquetés. Cette base de données comprend au moins deux colonnes : une contenant le texte du document et les autres représentant les étiquettes. La figure suivante illustre cette structure avec des documents et leurs étiquettes associées.

	text	LF	GC	GP	AP	GE	JM
1	ماسو الاحزاب الشيوعيه الوطن العربي ماسي الاحزاب الشيو	0	1	0	0	0	0
2	المامره الخفيه الاسلحه الفاسده تسلح الجيوش العربيه الو	0	0	0	1	1	0
3	حكم الانترنت بروتوكولات حكماا صهيون الفوضي الخلاقه ب	0	1	0	1	0	0
4	نجح الامريكيون بالفعل يضحكوا مسلم وعربي نجح الامريكيو	0	0	0	1	0	0
5	تقرير سري الخوف عصر جليدي يزيد قدمت وزاره الخارجيه	0	0	1	0	0	0

Figure 5 : Structure d'une base de données multi-étiquettée.

3.3.2 Nettoyage et prétraitement de données

Avant d'exploiter les données extraites pour l'objectif de notre projet, elles doivent passer par les étapes de prétraitement qui les rendent exploitable pour les algorithmes de classification. Cela inclut en générale les fonctions suivantes :

- Supprimer les « hashtags ».
- Enlever les espaces supplémentaires.
- Supprimer les Stopwords : comme " ثم " ,"في"
- Supprimer la ponctuation arabe et non arabe.
- Supprimer les caractères spéciaux.
- Supprimer les URLs.
- Supprimer les Emojis.

- Supprimer les mentions (@).
- Enlever « التشكيل ».
- Enlever « التطويل » : supprimer l'extension de ligne entre les lettres arabes.
- Normaliser la ligature : le remplacement de " ء " par " ئ" .
- Normaliser la « hamza » : le remplacement de la lettre finale " ة " par " ه " et la lettre " ى " avec " ي " finale.

3.3.3 Visualisation

Le système proposé permet les fonctions suivantes pour la visualisation des données :

- **Nuages de mots (WordCloud)** : Génère des nuages de mots pour représenter visuellement les termes les plus fréquents dans le jeu de données.
- **Visualisation des n-grammes** : Crée des représentations visuelles des bi-grammes, tri-grammes et n-grammes de plus haut niveau pour mettre en évidence les phrases courantes.
- **Graphiques interactifs** : Fournit des graphiques interactifs pour des insights plus approfondis sur l'affichage des données et l'évaluation des modèles de classification.

3.3.4 Vectorisation du texte

Cette fonctionnalité permet de transformer les données textuelles en vecteurs TF-IDF pour un entraînement de modèle classification efficace.

TF-IDF est une mesure statistique numérique utilisée pour évaluer l'importance d'un mot (terme) dans n'importe quel contenu d'une collection de documents en fonction des occurrences de tous les mots, et analyse également le niveau de pertinence des mots-clés utilisés dans le contenu donné. Elle considère non seulement la fréquence mais induit également une information discriminante pour chaque terme [21].

Le TF-IDF est donné par la formule suivante :

$$(TF\_IDF)_{i,j} = TF_{I,J} * IDF_{I,J}$$

Tel que :

$$TF_{I,J} = \frac{nombre\ d'occurrences\ de\ mot\ dans\ les\ documents}{nombre\ de\ mots\ dans\ tous\ les\ documents}$$

$$IDF_{I,J} = \log \frac{nombre\ de\ documents}{nombre\ de\ documents\ contenant\ le\ mot}$$

3.3.5 Catégorisation des documents

❖ Répartition des données

Une fois les textes numérisés, l'étape suivante est de diviser l'ensemble de données en deux parties : données d'apprentissage (entraînement) et données de test. Les données d'apprentissage sont utilisées pour entraîner les modèles de classification. Elles devraient représenter une grande partie de données disponibles. Les données de test sont utilisées pour évaluer la performance des classificateurs une fois qu'ils ont été entraîné . Idéalement, ces données devraient provenir d'une distribution similaire à celle des données d'entraînement, mais ne doivent pas être utilisées dans le processus d'entraînement.

Une fois que l'ensemble de données est divisé en ensembles d'entraînement et de test, on peut entraîner le modèles de classification sur les données d'entraînement et évaluer sa performance en utilisant les données de test. Cela permet de comprendre comment le modèle se comporte sur des exemples qu'il n'a jamais vus auparavant, ce qui est un indicateur crucial de sa capacité à généraliser à de nouvelles données.

❖ Transformation du problème de classification multi-label

Notre système de catégorisation multi-thématique des documents implémente trois approches pour transformer le problème de classification multi-label : Pertinence Binaire (Binary Relevance : BR), Ensemble de Puissances d'Étiquettes (Label Powerset: LP) et Chaîne de Classificateurs (Classifier Chain: CC). Les algorithmes illustrant ces approches sont indiqués ci-dessous

Algorithme 1 : Pertinence Binaire (BR)

1. Entrée :
- Un ensemble de données d'entrainement multi-étiquettes $D = \{(X_1, Y_1), (X_2, Y_2), .., (X_n, Y_n)\}$ où X_i est un document et $Y_i \subseteq \{1,2, \ldots, L\}$ est l'ensemble des étiquettes associées au document X_i.
- L est le nombre total d'étiquettes possibles.

2. Initialisation :
- Pour chaque étiquette I $(o\grave{u}\ I = 1,2, .., L)$, créer un classificateur binaire indépendant C_I

3. Entrainement :
- Pour chaque étiquette *I*:
- Construire un nouvel ensemble de données d'entrainement $D_I =$ $((x_1, y_1^I). (x_2, y_2^I), \ldots, (x_n, y_n^I)\}$ ou y_i^I est 1 si $I \in Y_i$ et 0 sinon.
- Entrainer le classificateur C_I sur D_I.

4. Prédiction :
- Pour un nouvel exemple x :
  - Pour chaque étiquette l:
    - Utiliser le classificateur C_l pour prédire si l doit être attribuée à x.
    - Combiner les prédictions de tous les classificateurs pour obtenir l'ensemble des étiquettes prédites $Y' \in \{1,2, \ldots, L\}$.

Algorithme 2 : Ensemble de Puissances des Etiquettes (LP)

1. Entrée :
- Un ensemble de données d'entraînement multi-étiquettes, $D = \{(X_1, Y_1), (X_2, Y_2), .., (X_n, Y_n)\}$

où chaque X_i est un document et $Y_i \subseteq \{1,2, \ldots, L\}$ est l'ensemble des étiquettes associées au document X_i.
- L représente le nombre total d'étiquettes possibles.

2. Transformation :
- Construire un nouvel ensemble de données d'entraînement où chaque combinaison unique d'étiquettes dans Y_i est traitée comme une étiquette unique.
- Par exemple, si $Y_1 = \{1,3\}\ et\ Y_2 = \{2\}$, ces combinaisons seront traitées comme de nouvelles étiquettes uniques.

3. Entrainement :
- Entrainer un classificateur multi-classes unique sur ce nouvel ensemble de données transformé.

4. Prédiction :
- Pour un nouveau document X:
  - Utiliser le classificateur multi-classes pour prédire l'étiquette unique correspondante
  - Décomposer cette étiquette unique en son ensemble original d'étiquettes multiples.

Algorithme 3 : Chaîne de Classificateurs (CC)

1. Entrée :
- Un ensemble de données d'entrainement multi-étiquettes $D = \{(X_1, Y_1), (X_2, Y_2), .., (X_n, Y_n)\}$,où chaque X_i est un document et $Y_i \subseteq \{1,2, \ldots, L\}$ est l'ensemble des étiquettes associées au document X_i.
- L represente le nombre total d ebquettes possibles

2. Initialisation :
- Créer une séquence ordonnée de L classificateurs $C_1, C_2, \ldots, C_l$

3. Entrainement :
- Pour chaque étiquette *l* de 1 a *L*.
  - Construire un ensemble de données d'entrainement D_l, en utilisant les caractenstiques onginales de X_i, et les étiquettes des classificateurs précédents comme caractéristiques supplémentaires.
  - Entrainer le classificateur C_l sur D_l ou l'étiquette cible est y_i^l, cest-saire 1 si $l \in Y_i$ et 0 sinon.

4. Prédiction :
- Pour un nouveau document *X* :
  - Intaliser un vecteur de prédiction vide *Y*'.
  - Pour chaque étiquette *l* de 1 à *L*.
    - Utiliser *X* et les prédictions précédentes dans *Y'* comme entrée pour le classificateur C_l.
    - Ajouter la prédiction de C_l à *Y's*.

❖ Classification des documents

Notre système de catégorisation des documents met en œuvre quatre classificateurs d'apprentissage automatique :

- Naïve de Bayes
- Arbres de décision
- Forêts aléatoires décisionnelles
- K plus proche voisins (kNN)

Le système intègre également des approches d'apprentissage profond adaptées aux tâches de classification multi-label de texte :

- Réseaux Neuronaux Récurrents (RNN)

- Réseaux Neuronaux Convolutifs (CNN)

3.3.6 L'évaluation

Dans l'évaluation des modèles de classification de texte à plusieurs étiquettes, le choix des métriques appropriées est crucial pour comprendre la performance du modèle et son aptitude à généraliser à de nouvelles données.

Notre système implémente des métriques telles que la précision, le rappel, la matrice de confusion, la F-mesure et l'exactitude offrent différentes perspectives sur la qualité des prédictions du modèle.

- **Précision :** (en anglais précision) : est définie comme le nombre de prédictions faites qui sont réellement correctes ou pertinent parmi toutes les prédictions basées sur la classe positive. Ceci est également connu comme valeur prédictive positive et peut être représentée par la formule [23][22] :

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

- **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 [23][22] :

$$\boldsymbol{Rappel} = \frac{nombres\ de\ vrais\ positifs}{nombres\ de\ vrais\ positifs + nombres\ de\ faux\ nigatifs}$$

- **F-mesure :** est une autre mesure de précision qui est calculée en prenant la moyenne harmonique de la précision et du rappel et peut être représentée comme suit [22] :

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

- **Taux de succès ou d'erreur ou exactitude :** (en anglais accuracy) : désigne le taux des prédictions réussites obtenu par le modèle de classification. Voici l'équation pour calculer exactitude [22] :

$$\boldsymbol{Taux\ de\ succès\ ou\ d'erreur} = \frac{nombre\ de\ predictions\ correctes}{nombre\ total\ de\ predictions}$$

- **Hamming loss :** 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 [24] :

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

  - ✓ N : Nombre d'échantillons (documents, observations).
  - ✓ L : Nombre total de labels (classes) possibles.
  - ✓ $y_{i,j}$: Vérité terrain (true label) pour l'échantillon i et l'étiquette j .
  - ✓ $y'_{i,j}$: Prédiction pour l'échantillon i et l'étiquette j

3.4 Conclusion

Dans ce chapitre, nous avons décrit les principales fonctionnalités de notre système de catégorisation automatique des documents arabes. Ces fonctionnalités permettent de créer un modèle performant et précis pour la classification multi-thématique des documents, offrant ainsi une solution complète et robuste pour l'analyse et la gestion des données textuelles en langue arabe.

4 Chapitre 04 : Implémentation et Résultats

4.1 Introduction

Dans ce chapitre, nous présentons l'environnement de travail et les outils utilisés pour mettre en œuvre notre système de catégorisation multi-thématique de documents arabe, nous clôturons ensuite par une discussion sur les résultats finaux et leurs implications.

4.2 Environnement de travail et outils d'implémentation

4.2.1 Matériel

Tableau 1: Table de matériels

Caractéristiques	Poste de travail N°1	Poste de travail N°2
PC	MSI	DELL
Système d'exploitation	Windows 10	Windows 11
Processeur	Intel(R) Core (TM) i7-7700HQ CPU @ 2.80GHz 2.80 GHz	Intel(R) Core (TM) i5-5200U CPU @ 2.20GHz 2.20 GHz
RAM	16.00GO	8.00 GO
Type de système	SE 64 bits	SE 64 bits

4.2.2 Environnement de programmation

❖ **Langage de programmation**

Nous avons choisi Python, versions [3.11.5] et [3.12.2] pour implémenter notre système de catégorisation des documents. Python est un langage de programmation interprété, multi-paradigme et multiplateformes. Il est conçu pour optimiser la productivité des programmeurs en offrant des outils de haut niveau et une syntaxe simple à utiliser. Il est doté d'un typage

dynamique fort, d'une gestion automatique de la mémoire par ramasse-miettes et d'un système de gestion d'exceptions. Le langage Python est placé sous une licence libre et fonctionne sur la plupart des plates-formes informatiques.

❖ **Éditeur de code**

Spyder intègre des bibliothèques scientifiques populaires de Python telles que NumPy, Pandas, Matplotlib et SciPy. Les Data Scientists peuvent donc les utiliser de manière transparente pour simplifier. Au sein de cet environnement unifié, les utilisateurs peuvent facilement manipuler les données, réaliser des calculs scientifiques complexes et créer des visualisations percutantes [25].

❖ **Librairies et bibliothèques Python intégrés**

Figure 6 : Bibliothèques Python utilisées

- **Pandas :** Pandas 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 [26].

- **NLTK :** est une plateforme Python leader pour travailler sur les données du langage naturel. Elle offre des interfaces faciles à manipuler et plus de 50 corpus et ressources lexiques. Nltk offre également des librairies de prétraitement de données, de classification, segmentation, racinisation, et plus d'autres [27].

- **PyArabic :** une bibliothèque de langue arabe spécifique pour Python. Elle fournit des fonctions de base pour manipuler les lettres et le texte arabes, comme détecter les lettres arabes, les groupes et les caractéristiques de lettres arabes, supprimer les signes diacritiques, etc [28].

- **Scikit-learn :** est une bibliothèque libre Python destinée à l'apprentissage automatique. Elle comprend de nombreuses méthodes pour la classification, la régression, et le clustering. Elle est conçue pour s'harmoniser avec d'autres bibliothèques libres Python, notamment NumPy et SciPy [29].

- **Keras :** Keras est un framework open source d'apprentissage profond pour le Python, capable de s'exécuter sur TensorFlow. Keras a été développé pour permettre des expérimentations rapides avec les réseaux de neurones profonds. Il se distingue par son extensibilité, compréhensibilité, rapidité et notamment sa simplicité [30].

- **Matplotlib :** est une bibliothèque de visualisation 2D de données, conçue pour Python. Elle offre des possibilités variées de visualisations statiques, personnalisées et interactifs dans des différents formats [31].

4.3 Description d'application

La section suivante présente notre application développée pour la catégorisation automatique multi-thématique des documents. Chacune des figures suivantes montre les fonctionnalités principales du système avec toutes les options qu'il offre.

❖ Interface principale

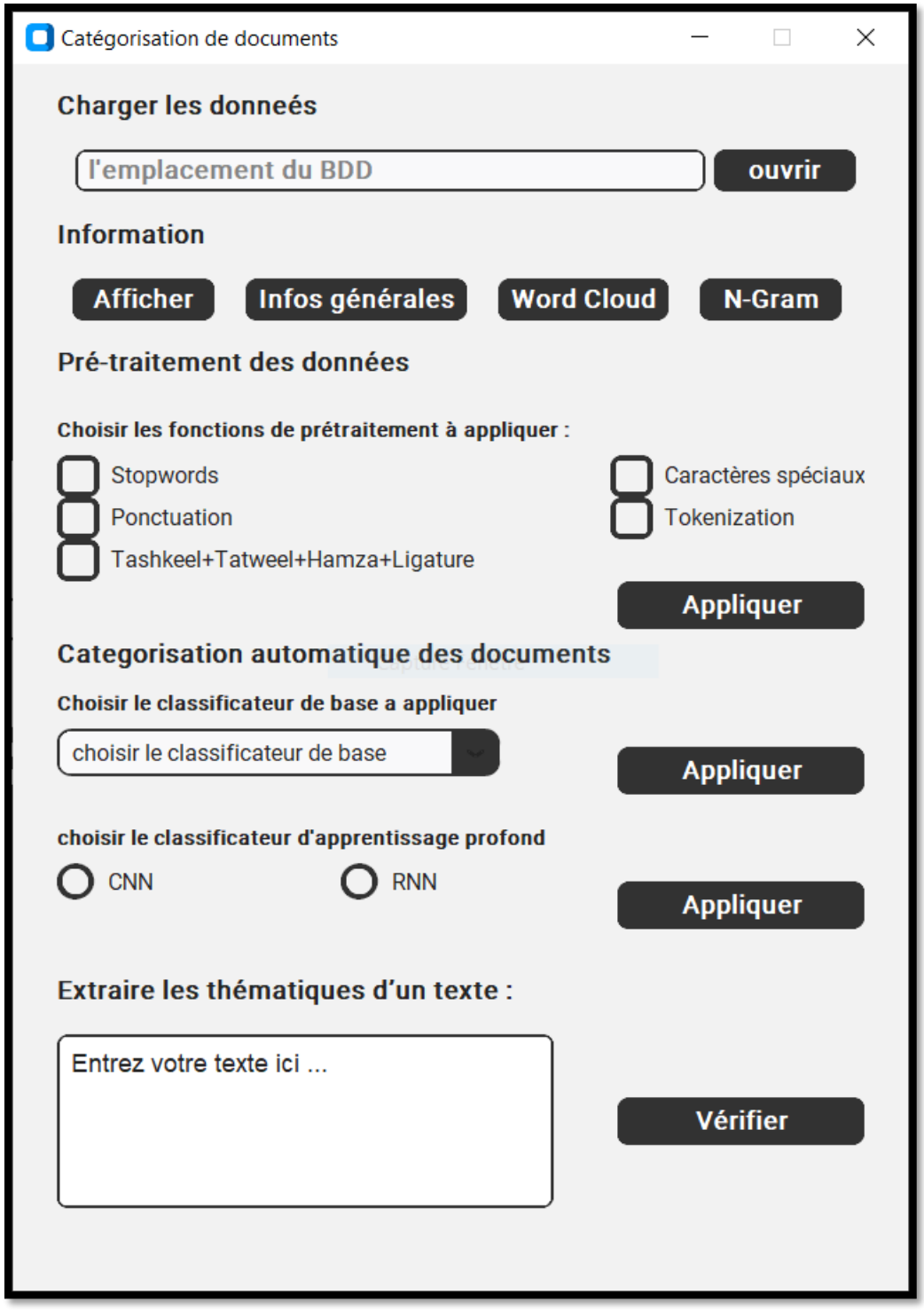

Figure 7 : Interface principale de l'application

❖ Chargement et visualisation de données

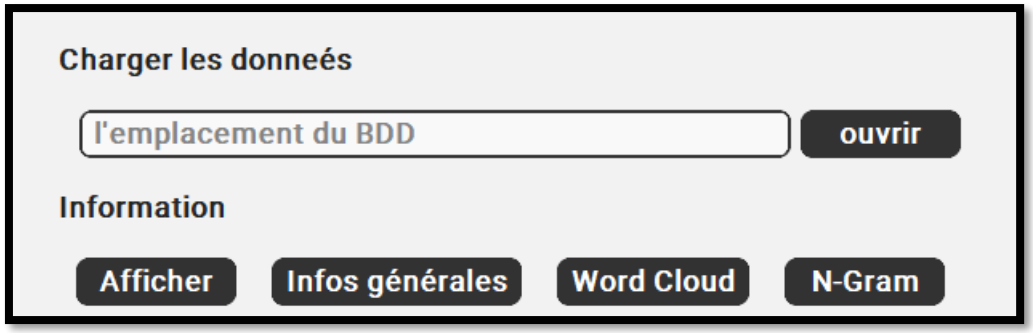

Figure 8 : Interface de chargement et visualisation des données

- ✓ **Bouton « Ouvrir »** : Ouvrir une fenêtre de dialogue pour sélectionner un fichier de base de données.
- ✓ **Bouton « Afficher »** : Affiche les données dans une nouvelle fenêtre.
- ✓ **Bouton « Infos générales »** : Capture les informations générales de l'ensemble de données et les affiche dans une boîte de message.
- ✓ **Bouton « WordCloud »** : Afficher le nuage de mots dans une nouvelle fenêtre
- ✓ **Bouton « N-Gram »** : Afficher les bigrammes et leurs fréquences, les mots et leurs fréquences, et créer un histogramme de la fréquence des mots dans une fenêtre Tkinter.

❖ Prétraitement de données

Figure 9 : Interface de prétraitement de données

- ✓ **Bouton « Appliquer »** : Appliquer les fonctions de prétraitement sélectionnées puis sauvegarder le résultat de prétraitement

❖ Catégorisation des documents

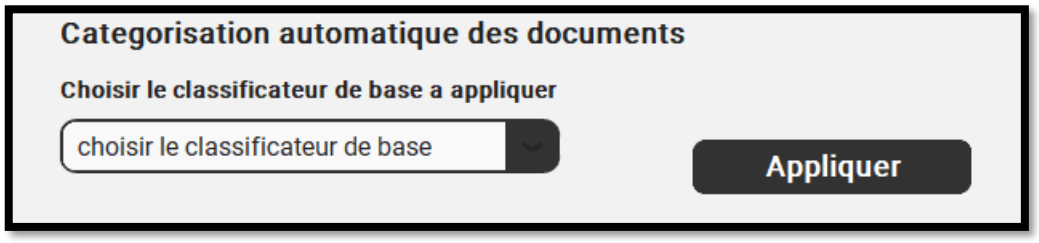

Figure 10 : Interface de catégorisation des documents à l'aide des classificateurs de base.

- ✓ **Bouton « Appliquer »** : Appliquer la méthode de classification sélectionnée à partir de la liste déroulante des classificateurs puis sauvegarder le résultat.

Figure 11 : Interface de catégorisation des documents à l'aide des modèles d'apprentissage profond

- ✓ **Bouton « Appliquer »** : Appliquer la méthode de classification sélectionnée à partir de la liste déroulante des classificateurs puis sauvegarder le résultat.

❖ Détection automatique des thématiques des documents

- ✓ **Espace de texte** : Zone de saisie de texte par utilisateur.

- ✓ **Bouton « Vérifier »** : Charger un modèle de classification pré-entraîné et sauvegardé de l'étape de catégorisation des documents pour détecter les thématiques d'un nouveau texte saisi par l'utilisateur.

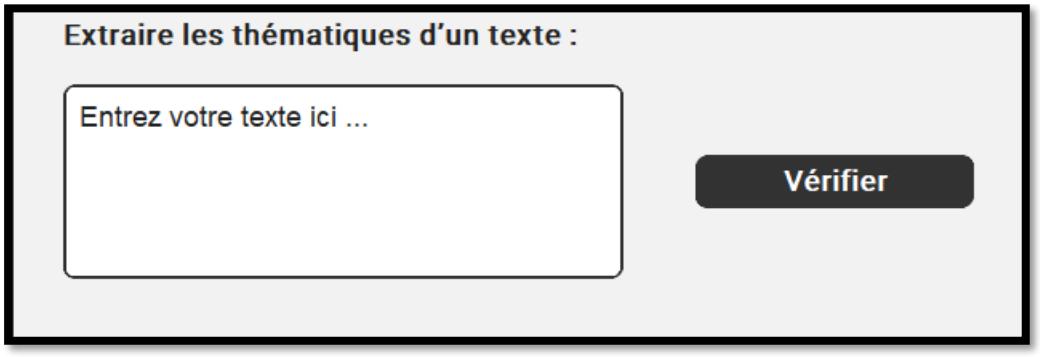

Figure 12 : Interface de détection automatique des thématiques des documents à l'aide des modèles déjà entraînés

4.4 Cas d'étude : Catégorisation des documents arabes

Dans cette section, nous présentons un cas d'étude en appliquant notre système de catégorisation de documents. Nous nous concentrons sur la catégorisation d'un ensemble de documents liés au sujet du conspirationnisme (i.e. théories de complot) extrait du contenu arabe publié en ligne sur le web et les réseaux sociaux. Ce cas d'étude permet de démontrer l'efficacité et la pertinence de notre application dans un contexte réel et d'actualité.

4.4.1 Description du jeu de données

L'ensemble de données comprend des textes en langue arabe (arabe standard et arabe dialectale) qui soulèvent et promeuvent divers sujets de théories du complot. Elle contient 2048 documents collectés à partir de 38 pages et recueillie dans le cadre d'un projet de fin d'études de Master intitulé « La modélisation thématique pour la caractérisation des fausses informations dans le contenu arabe en ligne » réalisé l'année passée 2023 par les deux étudiantes Guemraoui Zineb et Behih Dalila [31].

Les annotations de l'ensemble de données a été révisées et adaptées selon nos besoins. Par conséquence, les documents ont été catégorisé selon six classes selon la nature des théories de complots qu'ils incluent :

- **Le complot du genre/féministe [LF] :** Les croyances en la conspiration du genre impliquent l'idée que les activistes et les défenseurs du genre opèrent de manière secrète pour faire avancer un programme en conflit avec les normes sociétales établies.

- **Conspirations géopolitiques [GP] :** Les théories du complot géopolitique se concentrent sur les actions et les motivations des gouvernements, des organisations et d'autres acteurs sur la scène mondiale. Ces théories impliquent souvent des allégations de complots cachés, manipulatifs ou secrets opérant au sein de la dynamique politique mondiale ou régionale. Elles incluent souvent l'idée que certains événements sont orchestrés ou influencés par des organisations secrètes, des nations ou élites puissantes, ou des forces obscures cherchant à atteindre des objectifs spécifiques dissimulés au grand public.

- **Conspirations liées aux dissimulations gouvernementales [GC]** Les théories du complot liées à ce sujet pourraient impliquer des revendications de dissimulations gouvernementales sur la véritable nature de la Terre, l'existence de vie extraterrestre, la technologie secrète rétro-ingéniérée, ou des interactions cachées entre les humains et les êtres extraterrestres.

- **Apocalyptisme [AP]** Les croyances apocalyptiques, qu'elles soient religieuses ou séculaires, englobent une gamme diversifiée d'interprétations concernant des événements catastrophiques, souvent avec le potentiel de déclencher des scénarios de fin du monde.

- **La théorie du complot judéo-maçonnique [JM]** La théorie du complot judéo-maçonnique est une idéologie politique qui affirme qu'une alliance secrète de juifs et de francs-maçons complotent pour contrôler les événements et les institutions mondiales à des fins néfastes.

- **Théories du complot liées à la géo-ingénierie [GE]** Les théories du complot liées à la géo-ingénierie tournent autour de croyances spéculatives et souvent infondées selon lesquelles des entités puissantes, telles que des gouvernements ou des sociétés, sont secrètement engagées dans la manipulation à grande échelle de l'environnement terrestre.

Les autres caractéristiques de l'ensemble de données sont décrites ci-après.

Tableau 2: Caractéristiques de l'ensemble de données

Nombres de lignes	2048
Nombres des colonnes	7
Noms des colonnes	Text, Label 1[LF], Label 2[GC], Label 3[GP], Label 4[AP], Label 5[GE], Label 6[JM].
Vocabulaire	Le nombre de mots 525936
Catégories	• Le complot du genre/féministe • Conspirations géopolitiques • Conspirations liées aux dissimulations gouvernementales • Apocalyptisme. • La théorie du complot judéo-maçonnique • Théories du complot liées à la géo-ingénierie

4.4.2 Analyse exploratoire de données

❖ **Distribution des catégories**

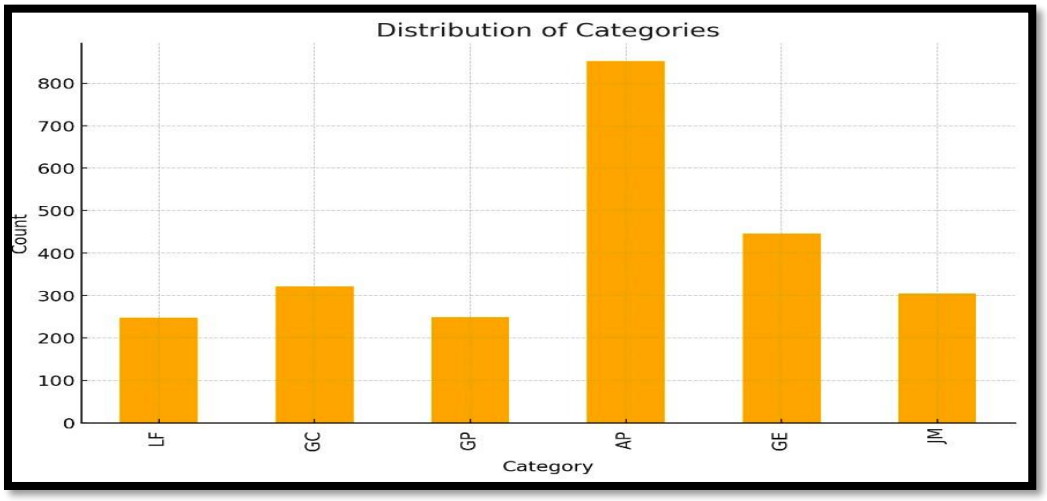

Figure 13 : Distribution des catégories.

Figure 13 représente un graphique à barres montrant la distribution des étiquettes pour notre ensemble de données. Notamment, la catégorie AP a le nombre le plus élevé, nettement plus haut que les autres barres. En revanche, les autres catégories affichent des chiffres inférieurs. Cette distribution asymétrique suggère que l'ensemble de données est déséquilibré, avec une catégorie dominante.

❖ Nuage de mots (WordCloud)

Le nuage de mots (en anglais, Word Cloud) est une technique qui nous permet d'afficher les mots les plus courants dans un corpus de textes donné. La figure ci-dessous présente le nuage de mots de notre jeu de données, où plus les mots sont fréquents, plus leur taille de police est grande [31].

Figure 14 : Le nuage de mots du jeu de données

Nous remarquons que le nuage de mots extrait reflète la diversité des concepts et des sujets dans la base de données étudiée. Les mots les plus fréquents apparaissent comme "المراة " , "

الامر". La présence de ces mots indique que les sujets des débats portent sur le complot du genre/féministe et les problèmes d'association. Il existe également un groupe de mots associés à la sphère complot judéo-maçonnique, tels que " الماسونية" , " الحقيقة".

Dans l'ensemble, ce nuage de mots donne un aperçu des principaux concepts et sujets couverts par les documents du jeu de données et contribue à la compréhension des principales tendances et modèles dans les données identifiées.

❖ Les N-Grammes

Un N-gramme est une tranche de N caractères d'une chaîne. Bien que dans la littérature le terme puisse inclure la notion de tous ensemble concomitant de caractères dans une chaîne (par exemple, un N-gramme composé du premier et du troisième caractère d'un mot) [32].

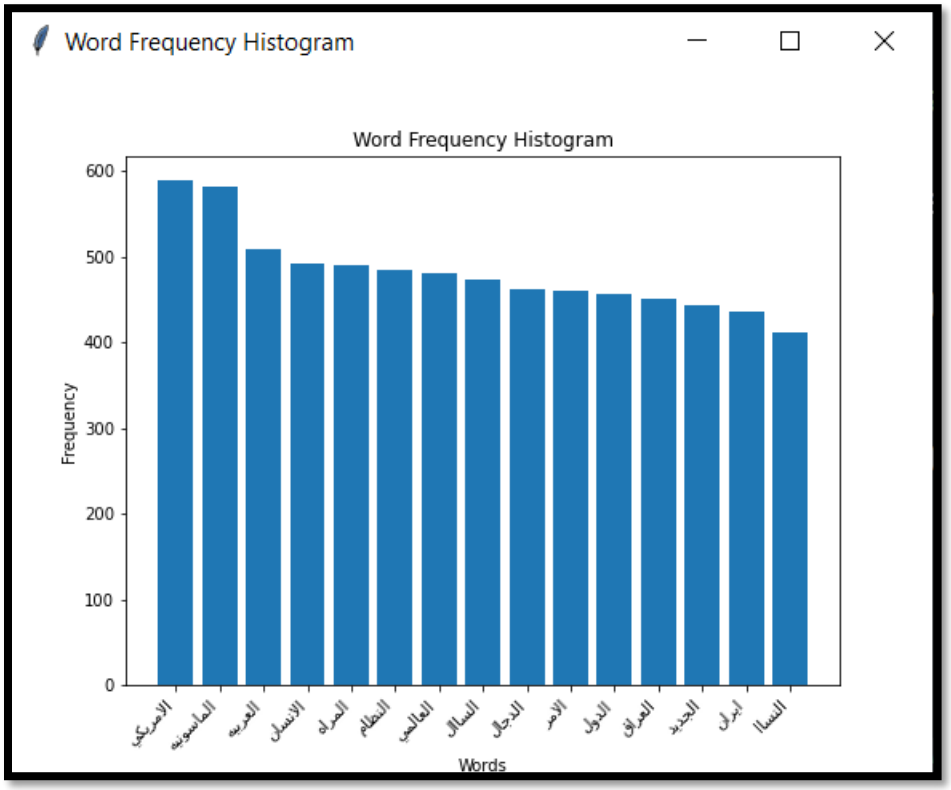

Figure 15 : Les 1-grammes les plus fréquents dans le jeu de données

Figure 16 : Les bi-grammes les plus fréquents dans le jeu de données

4.4.3 Prétraitement

Nous avons appliqué les fonctions de nettoyage et de prétraitement fournies par notre application à notre corpus. L'exemple suivant montre les textes avant et après le prétraitement :

Figure 17 : Texte avant prétraitement

Figure 18 : Texte après prétraitement

4.4.4 Vectorisation

Sur notre corpus, nous avons appliqué une vectorisation basée sur le TF-IDF, décrite dans le chapitre 3. L'exemple suivant montre les vecteurs d'un texte simple après la vectorisation :

دعوة للتفكر و التدبر نرى القطب الشمالي دااما اعلى صور النموذج الكروي و القطب الجنوبي اسفلها لمرة راينا صورة النموذج الكروي القطب الجنوبي اعلى الصورة و الشمالي اسفلها اليست السماا بزعمهم تحيط بالارض النواحي لماذا يصورون القطبين دااما احدهما اعلى الصور والآخر اسفلها اهي صدفة واية صدفة تستمر لقرونا

Figure 19 : Texte avant vectorisation

{دعوة: 0.115}, {للتفكر: 0.115}, {التدبر: 0.115}, {نرى: 0.115}, {القطب: 0.344}, {الشمالي: 0.229}
{دااما: 0.229}, {اعلى: 0.344}, {صور: 0.115}, {النموذج: 0.229}, {الكروي: 0.229},{الجنوبي:0.229
{اسفلها: 0.344}, {لمرة: 0.115}, {راينا: 0.115}, {صورة: 0.115}, {الصورة: 0.115}, {اليست: 0.115}
{السماا: 0.115}, {بزعمهم: 0.115}, {تحيط: 0.115}, {بالارض: 0.115}, {النواحي: 0.115}, {لماذا: 0.115}
{يصورون: 0.115}, {القطبين: 0.115}, {احدهما: 0.115}, {الصور: 0.115}, {والآخر: 0.115}, {اهي: 0.115}
{صدفة: 0.229}, {واية: 0.115}, {تستمر: 0.115}, {لقرونا: 0.115}

Figure 20 : Texte après vectorisation

Dans ce texte, les mots sont représentés par des vecteurs TF-IDF qui capturent à la fois leur fréquence dans le document et leur importance dans le corpus. Les mots spécifiques à chaque document auront des valeurs élevées pour TF-IDF dans ce document et des valeurs faibles dans les autres.

4.4.5 Catégorisation automatique des documents arabes

Dans cette section, nous présentons les résultats de la catégorisation automatique des documents arabes en utilisant notre application. Nous évaluons les performances des classificateurs de base (apprentissage automatique) et des modèles de deep learning selon les trois approches de transformation du problème de classification multi-étiquettes : Binary Relevance, Label Powerset, et Classifier Chain. Les résultats sont appliqués à notre jeu de données, illustrant l'efficacité et la précision de chaque approche pour la catégorisation multi-thématique des documents.

❖ Classificateurs de base

Les tableaux suivants représentent les résultats obtenus après avoir appliqué les classificateurs de base selon les trois approches de transformation.

Tableau 3 : Performance des classificateurs de base avec l'approche BR

METHODE	PRECISION	RAPPEL	F-MESURE	Hamming Loss
Naïve de Bayes	0,71	0,17	0,27	0,19
Arbre de décision	0,58	0,47	0,52	0,18
Foret aléatoire	0,5	0,1	0,17	0,21
KNN	0,52	0,53	0,52	0,2

Tableau 4 : Performance des classificateurs de base avec l'approche LP

METHODE	PRECISION	RAPPEL	F-MESURE	Hamming Loss
Naïve de Bayes	0,54	0,43	0,48	0,19
Arbre de décision	0,39	0,47	0,42	0,26
Foret aléatoire	0,36	0,3	0,33	0,26
KNN	0,48	0,47	0,47	0,22

Tableau 5 : Performance des classificateurs de base avec l'approche CC

METHODE	PRECISION	RAPPEL	F-MESURE	Hamming Loss
Naïve de Bayes	0,71	0,17	0,27	0,19
Arbre de décision	0,54	0,55	0,54	0,18
Foret aléatoire	0,77	0,20	0,32	0,17
KNN	0,75	0,40	0,29	0,16

❖ Méthodes d'apprentissage profond

Tableau 6 : Performance des classificateurs d'apprentissage profond

METHODE	PRECISION	RAPPEL	F-MESURE	Hamming Loss
CNN	0.73	0.41	0.52	0.14
RNN	0.99	1.0	0.99	0.006

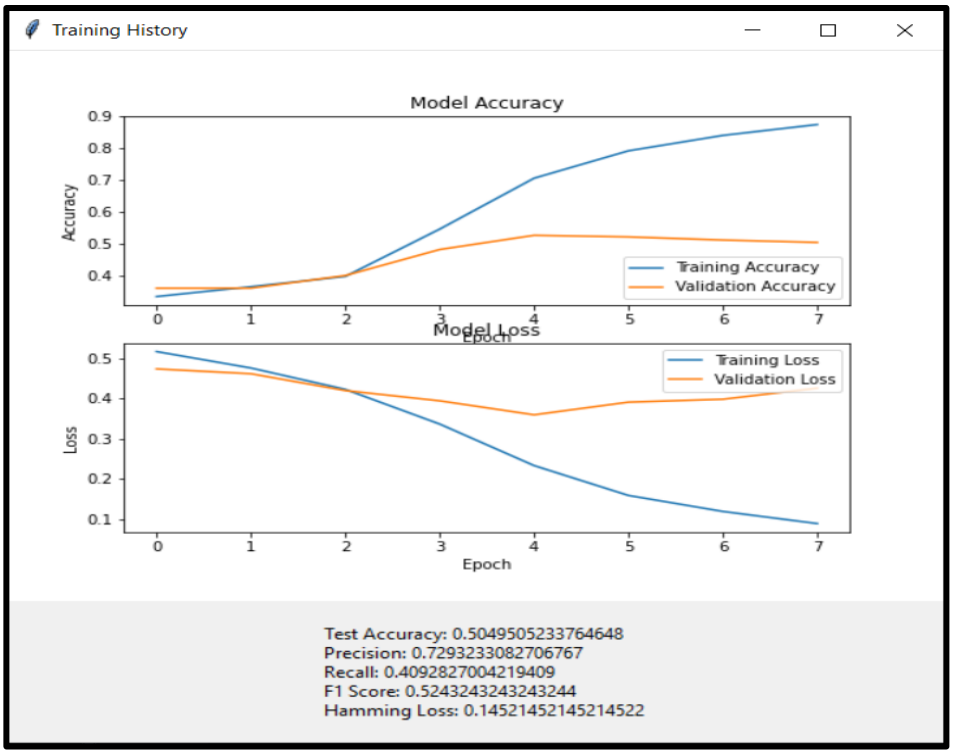

Figure 21 : Exactitude et perte de données selon les époques pour la méthode CNN

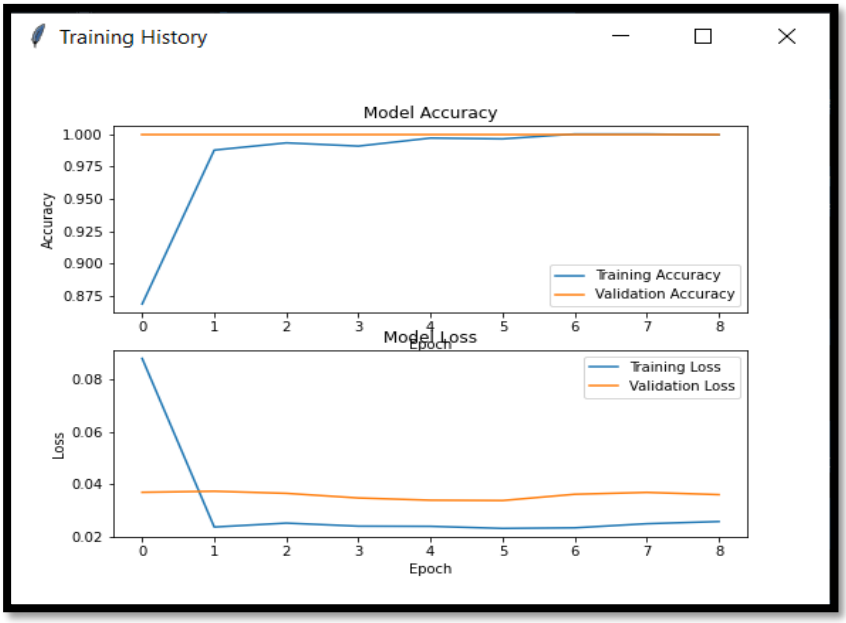

Figure 22: Exactitude et perte de données selon les époques pour la méthode RNN

4.4.6 Discussion des résultats

La comparaison entre les classificateurs de base et les méthodes d'apprentissage profond pour la classification multi-étiquettes de sujets sur notre ensemble de documents arabes révèle des informations significatives sur leurs performances à travers différentes métriques, mettant en évidence les forces et les limites de chaque méthode.

❖ **Classificateurs de base**

✧ **Approche de pertinence binaire (BR)**

La haute précision mais le faible rappel suggèrent que Naïve Bayes était très conservateur dans l'attribution des étiquettes, probablement en raison de son hypothèse d'indépendance des caractéristiques, qui ne se vérifie pas bien pour les données textuelles où le contexte est important.

Les Arbres de Décision, avec une performance plus équilibrée, ont mieux géré les dépendances entre étiquettes, mais leur tendance à surajuster pourrait avoir contribué à une perte de Hamming légèrement plus élevée.

La performance la plus faible indique que les Forêts Aléatoires, bien que généralement robustes, pourraient avoir eu du mal avec la parcimonie et la haute dimensionnalité de l'espace des caractéristiques dans les données textuelles.

KNN a atteint une performance relativement équilibrée suggérant que son apprentissage basé sur les instances était efficace dans une certaine mesure, bien que coûteux en calcul et sensible au choix de 'k' et de la métrique de distance.

✧ **Approche d’ensemble de puissances des étiquettes (LP)**

La diminution de la performance indique que Naïve Bayes a eu du mal avec la complexité introduite par le traitement de chaque combinaison d'étiquettes comme une seule classe, ce qui a probablement conduit à des problèmes de parcimonie des données.

Le déclin de la performance des Arbres de Décision et des Forêts Aléatoires reflète également la difficulté accrue de modéliser toutes les combinaisons possibles d'étiquettes et la propension à surajuster avec des données limitées pour chaque combinaison.

KNN a maintenu une performance relativement équilibrée, bien que sa sensibilité au choix des voisins et la malédiction de la dimensionnalité aient été des facteurs limitants.

✧ **Approche de chaînes de classificateurs (CC)**

Naïve Bayes a marqué une performance cohérente avec l'approche CC, indiquant que la modélisation des dépendances séquentielles n'a pas affecté de manière significative sa performance. Pour les Arbres de Décision, la plus haute F-mesure (0.54) parmi les classificateurs de base suggère que CC a aidé à capturer les dépendances entre étiquettes mieux que BR et LP.

Les Forêts Aléatoires ont marqué une amélioration de la précision mais toujours un rappel faible indiquent que, bien que CC ait aidé quelque peu, la complexité du modèle et la nature des données ont limité son efficacité globale.

De l'autre côté, KNN a atteint la meilleure perte de Hamming parmi les classificateurs de base, indiquant que la modélisation des dépendances d'étiquettes de manière séquentielle a bénéficié à KNN, bien que sa F-mesure globale soit modérée.

❖ **Méthode d'apprentissage profond**

La performance des méthodes d'apprentissage profond, spécifiquement les Réseaux de Neurones Convolutionnels (CNN) et les Réseaux de Neurones Récurrents (RNN), a été évaluée, révélant un contraste marqué par rapport aux classificateurs de base.

Le modèle CNN a montré une précision raisonnable de 0.73 et un rappel de 0.41, résultant en une F-mesure de 0.52 et une perte de Hamming de 0.14. Les CNN sont efficaces pour capturer des motifs locaux et des hiérarchies spatiales dans les données textuelles via des couches convolutionnelles. Cependant, leur capacité limitée à capturer des dépendances à long terme dans les données séquentielles a pu contraindre leur rappel, car les CNN excellent généralement dans

les tâches avec des caractéristiques locales fortes mais peinent avec les dépendances à long terme sans mécanismes additionnels comme l'attention.

D’autre part, le modèle RNN a significativement surpassé toutes les autres méthodes avec des valeurs de précision et de rappel presque parfaites, résultant en une F-mesure exceptionnelle et une perte de Hamming extrêmement basse. En fait, les RNN sont intrinsèquement conçus pour gérer les données séquentielles, les rendant exceptionnellement adaptés aux tâches de traitement du langage naturel. Leur capacité à maintenir une mémoire des entrées précédentes leur permet de capturer le contexte et les dépendances sur de longues séquences, crucial pour classer correctement les sujets dans les textes arabes.

Cependant, les résultats exceptionnels du RNN pourraient être dus à un surajustement, surtout s'il existe une étiquette prédominante dans l'ensemble de données qui biaise les résultats (étiquette **Apocalyptisme [AP]**). Le modèle peut avoir appris à se fier excessivement à cette étiquette dominante, entraînant des mesures de précision et de rappel gonflées qui ne se généralisent pas bien aux données non vues.

En somme, les classificateurs de base, malgré leurs degrés d'efficacité variables en fonction de l'approche multi-étiquettes utilisée, ont généralement eu du mal à équilibrer précision et rappel, menant à des F-mesures modérées et des valeurs de perte de Hamming plus élevées. En revanche, les modèles d'apprentissage profond, en particulier le RNN, ont démontré des performances exceptionnelles, soulignant l'importance de tirer parti des architectures de réseaux neuronaux avancés pour les tâches complexes de traitement du langage naturel impliquant des ensembles de données nuancés et riches en contexte comme le texte arabe.

Cependant, les résultats remarquables du RNN doivent être interprétés avec prudence. La possibilité de surajustement, en particulier en raison d'une étiquette prédominante dans l'ensemble de données, suggère que des validations supplémentaires sont nécessaires pour s'assurer que les performances du modèle reflètent réellement sa capacité à se généraliser plutôt qu'un artefact de la distribution des étiquettes de l'ensemble de données. Des techniques comme la validation croisée, l'utilisation d'un ensemble de données plus équilibré ou l'incorporation de

méthodes de régularisation pourraient fournir des informations plus robustes sur les performances réelles du modèle. Cela met en évidence le besoin crucial d'une évaluation et d'une validation approfondies lors du déploiement de modèles d'apprentissage profond dans des applications réelles.

4.5 Conclusion

À travers cette étude, nous avons exploré l'application de la catégorisation des documents en nous basant sur des données variées. Nous avons détaillé chaque étape de mise en œuvre, en examinant les performances de différents modèles de classification. Enfin, nous avons discuté les résultats obtenus en utilisant des métriques appropriées, offrant une analyse approfondie des limites et des performances de ces modèles.

Conclusion générale

Ce mémoire a présenté notre projet de développement d'une application de catégorisation automatique des documents arabes, en mettant l'accent sur une approche multi-étiquettes. L'objectif principal de ce projet était de créer un système capable de classer efficacement des documents selon plusieurs thèmes, en tenant compte des particularités de la langue arabe.

L'application développée intègre un ensemble complet de fonctionnalités, incluant le chargement et le prétraitement des données, la visualisation des informations, la vectorisation des textes, et l'utilisation de diverses techniques de classification. Nous avons implémenté et testé trois approches principales de transformation pour la classification multi-étiquettes : Binary Relevance, Label Powerset et Classifier Chain. Ces approches ont été utilisées avec des classificateurs de base d'apprentissage automatique ainsi que des modèles de deep learning, permettant ainsi une comparaison approfondie de leurs performances.

Pour valider l'efficacité de notre système, nous avons appliqué notre solution à un cas d'étude concret : la catégorisation de documents en ligne relatifs aux théories de complot dans le contenu arabe. Ce jeu de données a été collecté dans le cadre d'un projet de fin d'études de Master réalisé l'année passée par deux étudiantes. Les résultats obtenus montrent que notre application peut classifier avec précision et pertinence les documents selon leurs thématiques respectives.

Cependant, notre travail présente certaines limitations. L'application peut encore être améliorée en intégrant des méthodes plus avancées de traitement du langage naturel et en optimisant les modèles de deep learning pour mieux gérer les grandes variétés de textes et d'étiquettes. De plus, l'expansion du jeu de données pourrait fournir une base plus solide pour l'entraînement et l'évaluation des modèles.

En conclusion, ce projet a réussi à démontrer la faisabilité et l'efficacité d'un système de catégorisation automatique des documents arabes, tout en soulignant les domaines potentiels d'amélioration pour des recherches futures.

Les références

1. Abrecy. (2022, March 15). Text Mining: Classification Automatique de textes. HeadMind Partners. https://www.headmind.com/fr/text-mining-classification-automatique-de-textes/.
2. Université de Sherbrooke. (n.d.). Classification des documents - Service des bibliothèques et archives. https://www.usherbrooke.ca/biblio/archives/classification-des-documents
3. Koch, G. (2006). Catégorisation automatique de documents manuscrits: Application aux courriers entrants. S.l: s.n.
4. Périé, J. (2024, February 26). La classification des documents: un domaine d'application pour l'IA. Redsen. https://www.redsen.com/data-management/la-classification-desdocuments-un-domaine-dapplication-pour-lia/
5. Cours, tutoriaux et travaux pratiques. (n.d.). Classification automatique de textes. https://www.mcours.net/cours/pdf/leilclic3/leilclic929.pdf
6. Benblal, Z., & Belouafi, F. (2015). Intégration d'un lemmatiseur arabe dans le cadre d'un système de recherche d'information (Mémoire de fin d'étude).
7. Fehri, H. (2012). Reconnaissance automatique des entités nommées arabes et leur traduction vers le français (Doctoral dissertation). Université de Franche-Comté; Université de Sfax.
8. Aljedani, N., Alotaibi, R., & Taileb, M. (2020). Multi-Label Arabic Text Classification: An Overview. International Journal of Advanced Computer Science and Applications, 11(10). https://doi.org/10.14569/ijacsa.2020.0111086
9. Fallah, H., Bellot, P., Bruno, E., & Murisasco, E. (2023). Exploitation des dépendances entre labels pour la classification de textes multi-labels par le biais de transformeurs. EGC 2023 -23ème conférence francophone sur l'extraction et la gestion des connaissances, Lyon, France. pp. 31-42. https://hal.archives-ouvertes.fr/hal-04111691
10. Bounemra, R. (2023). Réalisation d'un système de classification des apprenants à partir d'indicateurs d'évaluation de leur apprentissage. Université de Guelma. https://dspace.univ-guelma.dz
11. Kariuki, C. (2021, September 24). Multi-Label Classification with Scikit-MultiLearn. Webscale's Engineering Education. https://www.webscale.com/engineering-education/multi-label-classification-with-scikit-multilearn/
12. Robert, J. (2023, November 9). La régression logistique, qu'est-ce que c'est? DataScientest. https://datascientest.com/regression-logistique-quest-ce-que-cest

13. Mostefa-Chebra, M. B., ESI, M. S. B., & Derouaz, M. L. (n.d.). Mémoire de projet de fin d'étude. ResearchGate. https://www.researchgate.net

14. Breiman, L. (2001). Random forests. Machine Learning, 45, 5-32.

15. Scikit-learn. (n.d.). RandomForestClassifier. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

16. IBM. (n.d.). Qu’est-ce que le boosting? https://www.ibm.com/fr-fr/topics/boosting

17. Datascientest.com. (n.d.). Transformer Neural Network: Qu'est-ce que c'est? Comment ça fonctionne? https://datascientest.com/transformer-neural-network-tout-savoir

18. LEBIGDATA.FR. (2023, October 4). Réseau de neurones artificiels: qu’est-ce que c’est et à quoi ça sert? https://www.lebigdata.fr/reseau-de-neurones-artificiels-definition

19. Saidj, S. D. (2022). Techniques de NLP pour la détection des fausses nouvelles. Université de Tiaret. https://dspace.univ-tiaret.dz

20. Omar, A., Mahmoud, T. M., Abd-El-Hafeez, T., & Mahfouz, A. (2021). Multi-label Arabic text classification in online social networks. Information Systems, 100, 101785.

21. Contributeurs aux projets Wikimedia. (2024, April 10). TF-IDF. Wikipedia. https://fr.wikipedia.org/wiki/TF-IDF

22. MachineLearningMastery.com. (2020). Precision, Recall, and F-Measure for Imbalanced Classification. https://www.machinelearningmastery.com/precision-recall-and-f-measure-for-imbalanced-classification/

23. Wikipedia contributors. (2024, April 26). Precision and recall. Wikipedia. https://en.wikipedia.org/wiki/Precision_and_recall

24. Cross Validated. (n.d.). What is a Hamming Loss? will we consider it for an Imbalanced Binary classifier. Stack Exchange. https://stats.stackexchange.com/questions/336820/what-is-a-hamming-loss-will-we-consider-it-for-an-imbalanced-binary-classifier

25. Spyder IDE. (2023). Home — Spyder IDE. https://www.spyder-ide.org/

26. Lelong, L. (2024, May 3). L'essentiel à savoir sur Pandas: la bibliothèque Python. Jedha Bootcamp. https://www.jedha.co/formation-analyse-donnee/pandas-bibliotheques-python

27. Albalawi, R., Yeap, T. H., & Benyoucef, M. (2020). Using Topic Modeling Methods for Short-Text Data: A Comparative Analysis. Frontiers in Artificial Intelligence, 3. https://doi.org/10.3389/frai.2020.00042

28. Linuxscout. (n.d.). pyarabic/paper.md at master · linuxscout/pyarabic. GitHub. https://github.com/linuxscout/pyarabic/blob/master/paper.md

29. Scikit-learn. (2024, April 17). https://en.wikipedia.org/wiki/Scikit-learn

30. Keras. (n.d.). Keras: Deep Learning for humans. https://keras.io/

31. Guemraoui, Z., & Behih, D. (2023). La Modélisation Thématique Pour La Caractérisation Des Fausses Informations Sur Les Médias Sociaux (Mémoire de Master). Université Mohamed El Bachir El Ibrahimi de Borj Bou Arréridj.

32. Cavnar, W. B., & Trenkle, J. M. (1994, April). N-gram-based text categorization. In Proceedings of SDAIR-94, 3rd annual symposium on document analysis and information retrieval (Vol. 161175, p. 14).

33. Boulaknadel, S. (2008). Traitement Automatique des Langues et Recherche d'Information en langue arabe dans un domaine de spécialité: Apport des connaissances morphologiques et syntaxiques pour l'indexation (Doctoral dissertation, Université de Nantes).

34. Benhadj Amar, M. R. (2021). Vers un système d'aide à l'automatisation des réponses aux requêtes de Fatwas islamiques. Université de Khemis Miliana. https://dspace.univ-km.dz

35. Kariuki, C. (2021, September 24). Multi-Label Classification with Scikit-MultiLearn. Webscale's Engineering Education. https://www.webscale.com/engineering-education/multi-label-classification-with-scikit-multilearn/

36. George, S., & Vasudevan, S. (2020). Comparison of LDA and NMF topic modeling techniques for restaurant reviews. Indian Journal of Natural Sciences, 10(62), 28210-28216.

37. Bouhali, W., & Ammara, B. (2022). La modélisation thématique pour le texte arabe (Mémoire de Master). Université Mohamed El Bachir El Ibrahimi-Bordj Bou Arréridj Faculté des Mathématiques et d'Informatique.