الجمهورية الجزائرية الديمقراطية الشعبية
The People's Democratic Republic of Algeria
وزارة التعليم العالي والبحث العلمي
Ministry of Higher Education and Scientific Research
جامعة محمد بوضياف بالمسيلة
University Mohamed Boudiaf of M'sila

كلية الرياضيات والإعلام الآلي
Faculty of Mathematics and Informatics

قسم الإعلام الآلي
Department of Computer Science

Domain : Mathematics and Computer Science

Thesis Presented to Fulfill the Partial Requirement
for **Master's Degree** in Computer Science

Specialty : Artificial Intelligence

Prepared By : Khouloud Arioua

Supervised By :
Dr : Bentercia Rahima

ENTITLED

Multi-label classification of Arabic text using deep learning models

Jury Members

Dr. ABDELBASSET BARKAT	President
Dr. BENTRCIA RAHIMA	Supervisor
Dr. BILAL LOUNNES	Examiner

Academic Year 2024/2025

DEDICATION

I dedicate this humble work to those whom God has commanded me to honor with righteousness and kindness, my beloved parents. May God prolong their lives and grant them continued health and well-being.

To those with whom we gathered one house, and they were the best support for my dear brothers, each in his name.

To my companions who left us and whose words remained in our ears, to those who taught me a letter throughout my academic path and were not stingy in giving it, my distinguished teachers, each in his name and position.

To myself, who bet on success, be patient and patient, for there is still a long way to go, and to all those to whom my heart has expanded and this paper has become too narrow to mention them, I dedicate my humble work to you, knowing that I am grateful to you, and in appreciation of your efforts.

ACKNOWLEDGMENT

First of all, I extend my sincere thanks and gratitude to God Almighty, for giving me the strength, wisdom and perseverance to accomplish this work.

I would like to express my sincere appreciation to :

My supervisor **Dr.Bentercia Rahima** for her valuable guidance and continuous support throughout the research journey.Her experience and patience played an essential role in completing this work.

Members of the thesis committee **Dr.Abdelbasset Barket** and **Dr.Bilal Lounnas** for their evaluation of my thesis.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

GENERAL INTRODUCTION

The evolution of data verification and formation is currently experiencing an unparalleled transformation. At the forefront of this change is Artificial Intelligence, which has emerged as a foundational discipline in the creation of intelligent systems capable of autonomous learning and decision-making. Consequently, through its approach to data, AI has been improving computational systems, enabling them to perform significantly more complex tasks with enhanced accuracy and efficiency.

Natural Language Processing (NLP) could very well be considered the most critical and fastest-changing branch of AI, with an interest in human language-machine interactions. With accelerating amounts of textual data in the world, particularly in diverse and linguistically rich languages like Arabic, there is an increasing need for automated systems for analyzing and understanding natural language truly well.

The idea of multi-label text classification emerges when a single document can be assigned to more than one relevant category. This is one of the atypical problems on the Arabic language as it includes complex morphological processes, syntactical ambiguity, and dialectal constructs. Traditional models cannot address these complexities, thus providing a vacancy for advanced models.

The emergence of deep learning techniques is an attempt to fill this gap with architectures that promise the capturing of semantic and contextual information from text data. Within this study, we concentrate on applying and evaluating deep learning models for multi-label classification of Arabic text, thereby contributing to the enhancement of Arabic language processing in modern intelligent systems.

Problem Statement

Notwithstanding the recent progress in natural language processing, particularly with the emergence of deep learning techniques, the domain of ”multi-label classification of Arabic texts” remains significantly underexplored. In contrast to single-label classification, multi-label classification introduces additional complexities, as a document may simultaneously belong to multiple categories. This is particularly relevant for Arabic texts that encompass a range of overlapping subjects, including politics, News, Sports, and Health. The present study addresses this gap by evaluating contemporary deep learning architectures, such as AraBERT and other transformer-based models, to develop an end-to-end system that can effectively execute multi-label classification on Arabic texts.

Motivation

While Arabic digital content is increasing, Arabic text remains underrepresented in the domain of multi-label classification. The complexity of language and the need to assign several categories to one document are often not effectively managed by existing systems. This situation prompted us to develop more sophisticated and accurate classification models, especially those that can process Arabic effectively using deep learning.

Objective

The primary objective of this research is to create and assess deep learning models for the multi-label classification of Arabic text. The specific goals are outlined as follows :

- To investigate the difficulties presented by the unique characteristics of the Arabic language within the framework of multi-label classification.
- To implement and compare various deep learning architectures, including QARiB, and MARBERT, and transformer-based models, particularly AraBERT, for the classification of Arabic text..
- To evaluate these models using suitable performance metrics that correspond to the multi-label nature of the task, such as Hamming Loss, F1-score, and accuracy.

Organization of The Manuscript

The structure of this manuscript is organized as follows :

Chapter 1: Multi-label classification of Arabic text

This chapter sets out a theoretical basis by presenting typical NLP concepts, reviewing the linguistic features of the Arabic language, touching upon the main classification paradigms,and the main challenges faced within Arabic NLP.

Chapter 2: related work

This chapter reviews existing research and collapses the state-of-the-art approaches under two categories : deep learning and machine learning methods.

Chapter 3: The Proposed Multi-label classification System

This chapter describes the adopted methodology for the study, classification system and details the tools, datasets, and implementation method used.

Chapter 4: Experimental Results

This chapter reports the results obtained using different deep learning models, comparing them through different evaluation metrics.

General Conclusion

In this final section, we give a summary of the research findings, limitations of the work, and possible directions for future work on Arabic multi-label text classification via deep learning.

CHAPTER 1

MULTI-LABEL CLASSIFICATION OF ARABIC TEXT

1.1 Introduction

Multi-label classification assigns multiple labels to each document simultaneously, which is suitable for Arabic texts that cover more than one topic. This chapter covers the theoretical aspects required to comprehend Arabic multi-label text classification deeply. It also discusses the special language features of Arabic, necessary preprocessing, and the differences among binary, multi-class, and multi-label classification. Next, it discusses some of the major challenges in Arabic Natural Language Processing (NLP) .

1.2 Natural Language Processing (NLP)

1.2.1 Definition

Natural Language Processing is a subset of artificial intelligence that centers on the interplay between computers and human language.This process revolves around the development of algorithms and models that are intended to empower machines to comprehend, analyze, and produce natural language text or speech [1] .

1.2.2 Natural Language Processing Stages

The NLP process is typically broken down into four distinct stages : text preprocessing, text representation, model training, and model evaluation. The core purpose of text preprocessing is to refine and prepare the raw text for further analysis [2] . Here are methods used in NLP :

1.2.2.1 Preprocessing

In this process, the focus is on refining and arranging unprocessed text data. Operations encompass tasks like diacritics removal, repetition reduction, stopwords elimination, letter forms normalization, and applying stemming or lemmatization techniques to simplify words to their root forms. Additional responsibilities, such as eliminating special characters and punctuation marks, along with the process of tokenization, contribute to organizing the text in preparation for subsequent analytical procedures.

1.2.2.2 Text Representation

After preprocessing, text needs to be converted into a numerical representation that is compatible with machine learning models.Common methods include :

- **Bag of Words :** A method that represents text by counting the frequency of each word in the document.
- **Term Frequency–Inverse Document Frequency :** Evaluates the significance of words by considering their frequency in various documents.
- **Word Embeddings :** such as Word2Vec, GloVe, and FastText are methods used to create compact vector representations that encode semantic connections among words.
- **Contextual Embeddings :** Sophisticated models such as BERT, AraBERT, and MARBERT provide flexible representations that adapt according to the context.

1.2.2.3 Modeling

This step entails the selection and training of a model to execute specific NLP tasks, such as text classification, sentiment analysis, or named entity recognition.Deep learning models such as Convolutional Neural Networks and Transformers have gained significant popularity for their capacity to analyze intricate patterns present in textual data.

1.2.2.4 Model Evaluation

After the training session, the model undergoes evaluation using suitable metrics to gauge its performance.In multi-label classification scenarios, Precision, Recall, F1-score, and Hamming Loss are utilized as evaluation metrics.These metrics give an indication of how effectively the model deals with instances containing overlapping or multiple categories.

1.3 Arabic Language Processing

1.3.1 Definition

Arabic, with over 300 million native speakers, is among the top five most widely spoken languages globally. It is also one of the six official languages of the United Nations. is renowned for its intricate morphology, which sets it apart significantly from IndoEuropean languages like English [3]. Arabic comprises 28 letters, including 25 consonants and 3 long vowels. Short vowels are not written as letters but as diacritical marks above or below the consonants. There are no lowercase or uppercase letters in Arabic ; all letters are of the same size. Arabic texts can be challenging to interpret due to their lack of vowels, leading to lexical ambiguity. The Arabic script is written from right to left, with letters taking different forms based on their position within a word - whether at the beginning, middle, or end end [4] .

1.3.2 Linguistic Characteristics of the Arabic Language

The Arabic language is known for its various unique linguistic properties that separate it from other languages and create notable complexities for natural language processing systems. The key elements encompass :

1.3.2.1 morphological

In Arabic, a word can signify a whole sentence thanks to its compound structure which is an agglutination of ”morphs” (roots, prefixes, affixes, suffixes, patterns). The following representation schematizes a possible structure of a word. It should be noted that reading and writing a word is done from right to left.

Tab. 1.1 : Structure of an Arabic word

Post fixed	Suffix	Schematic body	prefix	Antefix

- Antefixes are prepositions or conjunctions.
- Prefixes and suffixes express grammatical features and indicate functions : case of the noun, mood of the verb, and other categories of actualization (number, gender, person,...)
- Post fixes are personal pronouns [5] .

Example : The word **(أتتذكروننا)**

This word represents the English sentence : ”Do you remember us ?” When this word is broken down [4] , it consists of the following parts :

Tab. 1.2 : The segmentation of the word (أتتذكروننا)

Fixed post	Suffix	Schematic body	Prefix	Antefix
نا	ونـ	تذكر	تـ	ا
pronoun suffix complement of the name	verbal suffix expressing the plural	derived from the root	verbal prefix of the time of the unfulfilled	interrogation conjunction

1.3.2.2 Diacritics

Diacritical marks are signs added above or below Arabic letters to indicate the pronunciation of the word, this phonological role also influences the meaning of the word. There are three of these symbols transcribed as follows :

- Fatha ((الفتحة [a] is symbolized by a small line on the consonant (بَ ba)
- Damma (الضمة) [u] is symbolized by a hook above the consonant (بُ bu)

- Kasra (الكسرة) [i] is symbolized by a small line under the consonant (بِ bi).
- A small circle symbolizing sukun (سكون) is placed on a consonant when it is not linked to any vowel (بَعْدَ / baåda) [5] .

1.3.2.3 Word Order

In Arabic, the word that you wish to emphasize or the word that carries the most significance is positioned at the start of the sentence.This order causes artificial syntactic ambiguities in that the grammar must include all the rules for possible combinations of word order inversion in the sentence.Therefore, for instance, one could rearrange the words in a sentence (1) to form a different sentence (2) that holds the same meaning.

1. تعلم الطفل القراءة و الكتابة في المدرسة / the child has learned to read and write at school.
2. في المدرسة تعلم الطفل القراءة و الكتابة / at school the child has learned to read and write [6] .

1.3.3 Preprocessing Techniques for Arabic Text

Preprocessing is an essential component of Arabic NLP pipelines, tasked to convert raw text into a cleaner and more uniform typed text prepared for modeling. For Arabic, due to its nontriviality, it often could require some special purpose processing.

1.3.3.1 Normalization

Normalization is the process of providing relief from sparsity in your data by simplifying variants or letters and symbols. Common steps include :

- Mapping various forms of Alef (e.g. "آ" "إ" "أ") to a uniform .”ا”
- Regularizing Ta Marbuta (التاء المربوطة) in (ه) and(ت) context.
- Disregarding diacritics if any.

1.3.3.2 Tokenization

Tokenization is the process of breaking the text into meaningful subwords (or words). This is difficult in Arabic because of the script’s morphology and clitic ositioning. More advanced tokenizers such as Farasa, CAMeL Tools and MADAMIRA offer more accurate tokenization for Arabic.

1.3.3.3 Stop Word Removal

Commonly used Arabic stop words (e.g.”على “ ,”من”, “ في ” “) are semantically irrelevant, and have very limited semantic meaning and they are usually eliminated to help model in focusing on informative content. However, some stop words can be contextually relevant, their relevance will depend on the task at hand.

1.3.3.4 Stemming and Lemmatization

Stemming is the process of reducing words to their base form (e.g."كتب" <- "كتبت", "يكتبون") while lemmatization maps words to their lemma. These methods are useful for agglomerating words that have similar meaning and that have to be addressed separately, as well as for reducing sizes of vocabularies. Commonly used Arabic stemmers are Khoja and ISRI stemmers.

1.3.3.5 Non-Arabic Characters and Noise Processing

Arabic text derived from social media or web sources may contain latin characters, emojis, digits, punctuations,or long words (e.g."جمييييل"). Preprocessing often involves :

- Weeding out non-Arabic characters.
- Lessening the elongation of letter(e.g. "جميل" <- "جمممممميل").
- Punctuation and space .

1.4 Text classification

Text classification is the task of assigning documents to predefined categories based on their content.The process of automatically assigning natural language texts to predefined classes is known as automated classification. Text classification serves as the fundamental necessity for text retrieval systems, which are designed to fetch texts based on a user query, and text comprehension systems, which alter text through various means like generating summaries, addressing inquiries, or extracting information [7] . There are three primary paradigms of text classification based on the number and nature of class labels :

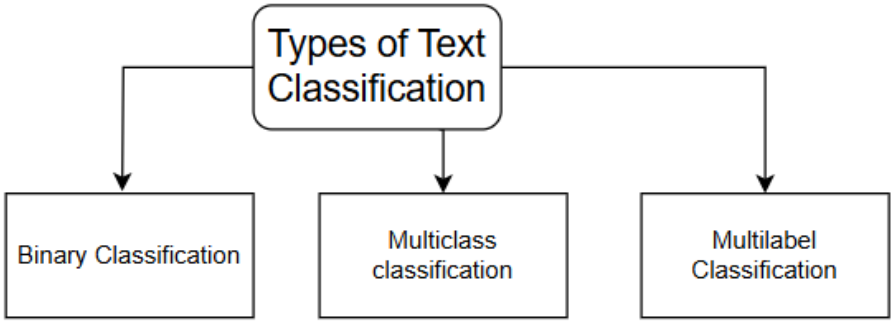

Fig. 1.1 : Types of text classification

1.4.1 Binary Classification

This problem is among the most basic classification tasks we may encounter. In a binary dataset, each instance is characterized by a solitary output attribute that can have only two possible values.These are typically labeled as positive and negative, yet they can alternatively be viewed as any other duo of values.One typical case of this task is spam filtering (see figure 1.2), where the classifier is educated based on the content of the messages to recognize those that are labeled as spam [8] .

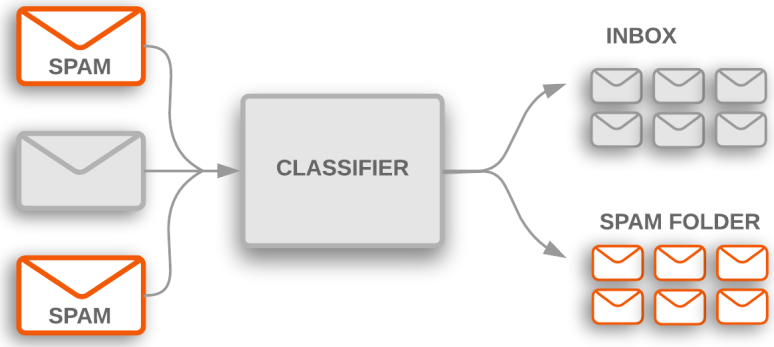

Fig. 1.2 : Binary Classification

1.4.2 Multi-Class Classification

Multi-class classification is the task of classifying a new instance into one among at least three classes. Multi-class classification tasks are prevalent across various fields, including image recognition, document categorization, gene expression analysis, and more.It is a well-established notion that the level of difficulty escalates in multi-class classification challenges as the quantity of classes grows, yet the specific reasons for this increased complexity remain a subject of inquiry [8] .

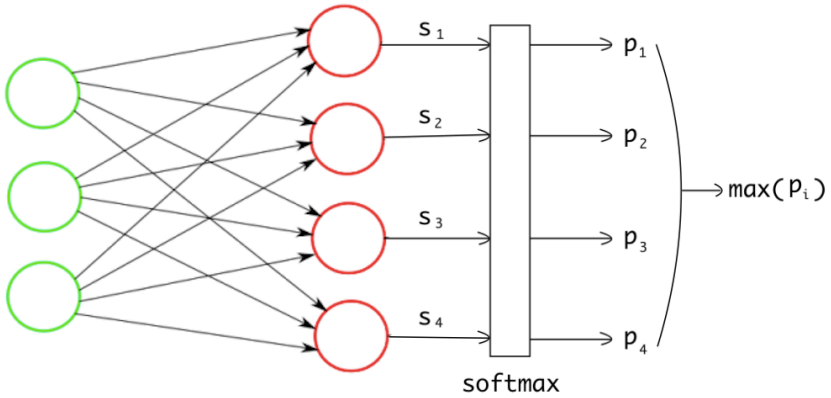

Fig. 1.3 : Multi-Class Classification

1.4.3 Multi-Label Classification

Multi-label text categorization pertains to the process of classifying a textual document into one or multiple categories. Text classification and multi-label text classification are frequently employed in contentious situations. For example, within the field of medical diagnostics, a patient may simultaneously present with both diabetes and stomach cancer; music classification; semantic scene analysis; and email spam identification. An image in semantic scene categorization can pertain to multiple conceptual categories simultaneously, for example, beaches and sunsets. In the same vein, a musical track could be characterized as being part of several genres within the scope of music classification [9] .

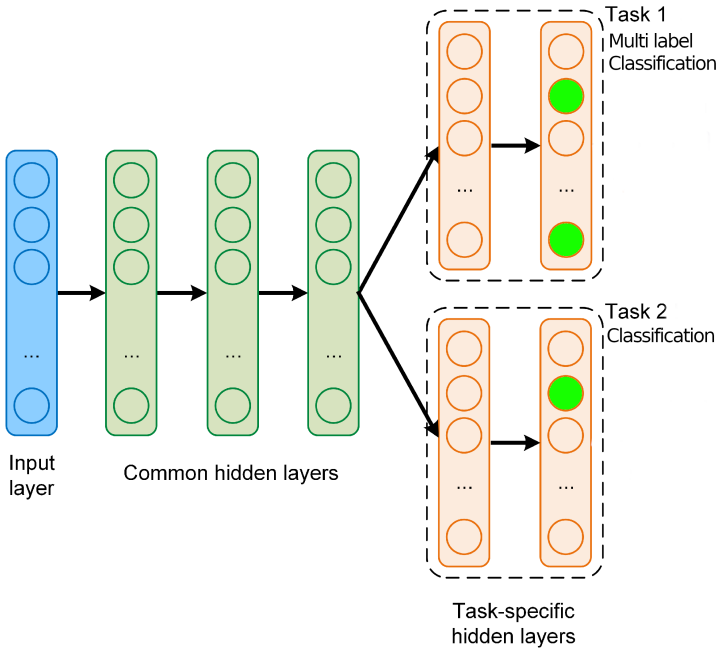

Fig. 1.4 : Multi-Label Classification

1.5 Challenges in Multi-Label Classification of Arabic Text

The multi-label classification of Arabic text presents many challenges because of the inherent complexity of the Arabic language and the technical constraints in current NLP tools and datasets .The following challenges can be succinctly outlined :

1.5.1 Scarcity of Labeled Multi-Label Datasets

While many Arabic datasets exist for single-label tasks , few are annotated for multi-label purposes.The scarcity of resources creates barriers to effectively training and evaluating deep learning models in multi-label contexts.

1.5.2 Linguistic Complexity of Arabic

The Arabic language is known for its rich morphology, making it a challenging language for natural language processing. With its intricate word formations that include prefixes, infixes, and suffixes, tasks like tokenization and stemming become more complex. Moreover, the absence of diacritics in written Arabic often results in ambiguity, where one word can have multiple meanings. The flexible word order in Arabic adds another layer of complexity to syntactic analysis and context understanding.

1.5.3 Imbalanced Label Distribution

In multi-label datasets, label imbalance is a frequent problem, with certain labels being more prevalent than others. This can cause models to exhibit a bias towards dominant classes, impacting their performance on less common topics such as ”Health” or ”Environment.”

1.5.4 Label Correlation and Dependency

Multi-label classification significantly expands the output space when compared to single-label tasks. This complexity requires more computational resources, especially when training deep learning models on extensive Arabic datasets.

1.6 Conclusion

In summary, this chapter thoroughly explained the theoretical and technical background of Arabic multi-label text classification. It explained the complicated features of the Arabic language, preprocessing methods suitable for Arabic text, and classified frameworks into binary,multi-class, or multi-label. Further, it looked into critical issues including the lack of datasets,the imbalance of labels, and dependency among labels. These insights set the ground for the methodology followed in the next chapters.

CHAPTER 2

RELATED WORK

2.1 Introduction

Artificial Intelligence (AI) is a branch of computer science that attempts to build systems that can perform tasks that require human intelligence, such as reasoning, learning, and decision-making. Natural Language Processing is a very apt domain under AI through which a machine is made to understand and generate human language. In recent years, there has been an unprecedented increase in Arabic online content, and thus, there is a growing interest in developing automated systems capable of efficiently understanding, categorizing, and processing Arabic texts. This chapter reviews the most relevant and up-to-date inquiries involving deep learning models, such as CNN, LSTM, GRU, and transformer-based architectures, along with machine learning models, including Logistic Regression, Random Forest, and Support Vector Machines (SVM). Each study is summarized concerning the dataset, features, what models used, and how well they performed overall. The reviewed studies provide overview insight into how data are preprocessed and performance trends that are relevant for Arabic multi-label classification tasks.

Fig. 2.1 : Overview of AI
[10]

2.2 Deep Learning Approaches

Deep learning approaches utilize multi-layer neural networks to detect complex features straight from text. The models can boast powerful performances in Arabic multi-label classification. Models such as CNN, LSTM, GRU, and ARABERT have given good results. In these reviewed studies, which are shown below in Table **2.1**, the most popular deep learning models were used. A good F1-score of 81.6 % and 83.8 % was achieved by **[11]** on a manually collected dataset from Arabic News and Mowjaz Multi-Topic. Another study also showed a good accuracy of 82.5% ,82.4% , and 82.5 % in its three models on the database **[12]**, while study **[13]** achieved excellent results with an accuracy of 94.85 % , 90.17 % , and 94.03 % in all its models. while the Study **[14]** had acceptable values, recording 78 % F1-Score , 77% ,78% , and 79% .In its four models, which were trained on a database taken from YouTube and Twitter. All these studies come together in one process, which is cleaning the data first before training it, which includes :(Remove extra spaces, Remove HTML tags, Remove Twitter usernames, web addresses, Separate punctuation from words and remove non-included marks, Remove underscores, Remove double quotes, Remove single quotes, Remove numbers, Remove stop words).

Tab. 2.1 : Sample of previous studies that belong to Deep Learning Approaches

Reference	Dataset	Models	Performance
[11]	Arabic News	Multilayer Perceptron (MLP)	81.6 % F1-score
	Mowjaz Multi-Topic	Long Short-Term Memory (LSTM)	83.8 % F1-score
[12]	Mowjaz Multi-Topic	AraBERT	82.5 % acc
		Asafya	82.4 % acc
		ARBERT	82.5 % acc
[13]	news websites	CNN-GRU (CGRU)	94.85 % acc
		LSTM	90.17 % acc
		BiLSTM	94.03 % acc
[14]	YouTube Twitter	CNN	78 % F1-Score
		BiLSTM	77% F1-Score
		BiGRU	78% F1-Score
		BERT	79 % F1-Score

2.3 Machine Learning Approaches

This method depends on conventional algorithms like Logistic Regression, Support Vector Machines, Random Forest, and XGBoost to identify patterns in labeled datasets.Feature extraction techniques like TF-IDF are widely employed, and multi-label classification is often managed using transformation methods such as OneVsRest.In the studies reviewed, as presented in Table **2.2**, various machine learning models were employed, yielding notably low accuracy in Study **[15]**, which utilized the Hana-MC database, resulting in accuracies of 46%, 27%, and 47%. Conversely, Study **[16]**, which analyzed the RTANews database, reported a very low Hamming score, indicating satisfactory model performance. Additionally, Study **[13]** demonstrated commendable accuracy, achieving rates of 81.34% and 84.73% across its models, while Study **[17]** attained an accuracy of 75% with the ISlamic database.

Tab. 2.2 : Sample of previous studies that belong to Machine Learning Approaches

Reference	Dataset	Feature extractor	Models	Performance
[15]	Hana-MC	TF-IDF	MLKNN	46% acc
			MLTSVM	27% acc
			BRKNN	47% acc
[16]	RTANews	TF-IDF	Logistic Regression	0.1 hammin loss
			Random Forest	0.09 hammin loss
			Multinomial Naive Bayes	0.1 hammin loss
[13]	news websites	TF-IDF	OneVsRest	81.34% acc
			OneVsRest+XGBoost	84.73 % acc
[17]	Islamic Fatwa	TF-IDF	LP-SVM	74.83% acc
			HOMER	75.8% acc
			Balanced k-means	75% acc

2.4 Conclusion

In summary, it can be stated from the above studies that, in many Arabic multi-label classification tasks, especially those involving big and well-preprocessed datasets, deep learning methods have the upper hand over the traditional machine learning methods. In some cases, models like CNN-GRU orAraBERT-based architectures even surpass the 90% threshold in term of accuracy. In contrast, traditional machine learning techniques can still cast a shadow if accompanied by good feature extraction and transformation.Besides, all studies accentuated data preprocessing as a must-do, including cleaning HTML tags, eliminating Twitter-specific ele-

ments, punctuation, and even stop words to help improve model performance. The review also shed light on machine learning being more interpretable and efficient in scenarios involving very small datasets, whereas deep learning models prove their advantage in robustness and scalability. Considering the highlighted strong and weak points of the aforementioned studies, the coming chapter proceeds to present the multi-label Arabic text classification .

CHAPTER 3

THE PROPOSED MULTI-LABEL CLASSIFICATION SYSTEM

3.1 Introduction

This chapter gives a multi-label text classification system for Arabic news articles, depending on a superior set of transformer-based models to address text categorization with multiple overlapping themes. The chapter elucidates the methodology and tools utilized and the strategies pursued in building and training the system, presenting a structured outline before analyzing its efficacy.

3.2 Work environment

3.2.1 Hardware

In this research, we relied on the Kaggle platform as an environment for training models. I benefited from the resources available on it, including : Two NVIDIA Tesla T4 GPUs,Up to 29 GB of RAM and Temporary storage space of up to 57.6 GB .

3.2.2 Software

In addition to the hardware , several software tools and platforms were employed to facilitate model development, training .

3.2.2.1 . Programming Language

- **Python :** Python is an object-oriented programming language with dynamic semantics, interpreted at a high level. Python's syntax, known for its simplicity and ease of comprehension, aims to prioritize readability as a means of minimizing the costs related to program maintenance. The Python interpreter, together with its expansive standard library, is offered in source or binary form at no charge for all major platforms, and is eligible for unrestricted distribution [18].

Fig. 3.1 : Python logo .

3.2.2.2 Libraries and frameworks

- **Pandas :** pandas is a Python library that offers efficient, adaptable, and articulate data structures crafted to facilitate the manipulation of ”relational” or ”labeled” data in a straightforward manner.Its goal is to be the core high-level module for carrying out hands-on, real-world data analysis in Python [19].

Fig. 3.2 : Pandas .

- **NumPy :** NumPy is the primary package for scientific computations in Python. It comprises a Python library that introduces a multidimensional array construct, various associated objects, and a plethora of algorithms tailored for efficient array manipulations [20].

Fig. 3.3 : Pandas .

- **Scikit-learn :** Scikit-learn represents an open-source machine learning library designed to facilitate both supervised and unsupervised learning techniques. It additionally furnishes a multitude of tools for model fitting, data preprocessing, model selection, model evaluation, and numerous other utilities [21].

Fig. 3.4 : Scikit-learn .

- **PyTorch** PyTorch is an open-source deep learning framework designed to offer flexibility and modularity for research purposes, while also ensuring the stability and support required for production deployment. It includes a Python package that facilitates high-level functionalities such as tensor computations, akin to NumPy, enhanced by robust GPU acceleration. Additionally, PyTorch features TorchScript, which simplifies the transition between eager execution and graph-based execution. The most recent version of PyTorch introduces capabilities such as graph-based execution, distributed training, mobile deployment, and quantization [22].

Fig. 3.5 : PyTorch .

- **Transformers (Hugging Face) :** Transformers represents a collection of pre-trained models in the fields of natural language processing, computer vision, audio processing, and multimodal tasks, designed for both inference and training purposes.Utilize Transformers to facilitate the training of models on your unique dataset, establish inference applications, and produce text leveraging extensive language models [23].

Fig. 3.6 : Transformers (Hugging Face) .

3.2.2.3 Development Environments

- **Kaggle :** A branch of Google, this platform serves as a digital hub for professionals specializing in data science and machine learning. provides a platform for individuals to locate datasets suitable for developing AI models, share their own datasets, engage in collaborations with other data scientists and machine learning experts, and take part in competitions

designed to tackle data science challenges. commenced its operations in 2010 by providing machine learning and data science competitions, in addition to furnishing a public data repository and a cloud-based business platform for data science and artificial intelligence education [24].

Fig. 3.7 : Kaggle logo .

3.2.2.4 Tools for Thesis Preparation

- **LaTeX (Overleaf) :** LaTeX is a tool for typesetting professional-looking documents. To produce a visible, typeset document, your LaTeX file is processed by a piece of software called a TeX engine which uses the commands embedded in your text file to guide and control the typesetting process, converting the LaTeX commands and document text into a professionally typeset PDF file. This means you only need to focus on the content of your document and the computer, via LaTeX commands and the TeX engine, will take care of the visual appearance [25].

  Overleaf An online collaborative LaTeX editor used for writing and formatting the Master's thesis, ensuring a professional academic presentation of the document.

Fig. 3.8 : LaTeX (Overleaf) .

3.3 Dataset Description

The database used in this project is derived from NADiA1, which includes Arabic text data collected through news websites using Python. It contains 35,416 files. It was in the form of txt [26], and we converted it to csv to be processed later to become 35,404 files(6814 single-label,28590 multi-label) divided into 24 categories displayed in English.

Labels : News, North Africa, Levant, Middle East, The Americas, Research, Finance & Economy, War & Terrorism, Gulf, Europe, Political Figures, Iran, Technology, Russia, Sports, Tennis, Football, English League, Arabian Sports, Spanish League, Health, East Asia, Environment, Other Countries.

	text	News	North Africa	Levant	Middle East	The Americas	Research	Finance & Economy	War & Terrorism	Gulf	...	Sports	Tennis	Football	English League	Arabian Sports	Spanis Leagu
0	وتتراوح الاحكام وخمس بتهمه دخول مجلس الامه صدر...	1	0	0	1	0	0	0	0	1	...	0	0	0	0	0	
1	اعلن المهاجم المنضم حديثا الفريق الكتالوني عثم...	0	0	0	0	0	0	0	0	0	...	1	0	1	0	0	
2	وتولي بنكيران رئاسه الحكومه اطار دستور جديد جر...	1	1	0	1	0	0	0	0	0	...	0	0	0	0	0	

Fig. 3.9 : Dataset NADiA1 .

3.3.1 Pre-Processing

In disciplines like natural language processing, the preprocessing of data is essential for enhancing the quality of the data. In the present project, the pyArabic package was employed for the purpose of data cleaning. Data cleaning refers to the procedure of rectifying and eliminating erroneous, corrupted, improperly formatted, or redundant data (see chapter1). The preprocessing phase encompasses the following stages :

- Remove extra spaces.
- Remove html tags.
- Remove twitter usernames, web addresses.
- Separate punctuation from words and remove not included marks.
- Remove underscores.
- Remove double quotes.
- Remove single quotes.
- Remove numbers.
- Remove Stopwords

3.3.2 Class Weights

To tackle issues related to class imbalance in the training set, we compute inverse class weights on the basis of the number of samples of each class. The weights are assigned in such a way that if the sample count is less for a certain class, then the class gets a higher weight. They will be calculated by dividing the number of samples by the product of the number of classes and the sample count of each class. The clipping of weight values within a lower bound of 0.7 and an upper bound of 4.0 is done to avoid any extreme values. These weights are given to the training process to pay more attention towards minority classes.

3.4 The Proposed Model

In this part, we will describe our research and the models we utilized. We implemented three deep learning models :ARABERT ,MARBERT, and QARIB Each model will be detailed in the following :

3.4.1 AraBERT

AraBERT is a transformer-based language model constructed by evolving the BERT architecture toward the Arabic language. Following the original BERT's training procedure, this model principally uses the Masked Language Modeling (MLM) task. It is pre-trained on a mammoth corpus of approximately 70 million Arabic sentences (almost 24GB of text) drawn from news articles, Wikipedia, and the Web. AraBERT comes in several variants : AraBERTv0.2-base, AraBERTv0.2-large, AraBERTv2-base, and AraBERTv2-large. The v2 models, in fact, are trained on pre-segmented text where affixes like prefixes and suffixes are split, a process that is central to handling complex Arabic morphology with much effectiveness. These models are released through Huggingface and implemented on PyTorch [12].

The fine-tuning in this work was carried out on the AraBERTv0.2-base model (aubmindlab/bert-base-arabertv02) by utilizing the AutoModelForSequenceClassification class of the Transformers toolkit. Input tokenized data were passed to the model with padding and truncation applied to a max length of 512 tokens to provide a wise trade-off between the performance and the efficient use of GPU resources. The classification head was modified to have 24 output neurons corresponding to all of the target categories, with sigmoid activations that allow multi-label output. Loss was computed via BCEWithLogitsLoss class-weighted variant inside a bespoke WeightedTrainer class, which is well suited to multi-label classification scenarios. This input was fed through a custom Dataset class in PyTorch that tokenized each input string and converted those tokens into input IDs and attention masks, along with converting the binary label vectors into PyTorch tensors. The Trainer class from Hugging Face was used for performing optimization steps, including gradient updates, evaluation on validation sets, checkpointing, and metric reporting during training. Metrics, including Hamming loss and accuracy, were computed by thresholding

the sigmoid outputs at 0.3 to convert outputs to binary predictions.

3.4.2 MARBERT

This model was specifically developed for the Arabic language ; however, in contrast to AraBERT, it underwent pretraining on an extensive dataset derived from Twitter, encompassing text in both Modern Standard Arabic and various Arabic dialects. The pretraining corpus consists of 1 billion tweets, amounting to nearly 128 gigabytes of textual data. The total token count within this corpus is approximately 15.6 billion, which is nearly twice the token count of Version 2 of AraBERT, thereby establishing it as the largest pretraining corpus among the nine models. MARBERT shares the same architecture as Multilingual BERT. The total number of trainable parameters in MARBERT is estimated to be around 160 million [27].

In this work, MARBERT was finetuned using the Transformers library's AutoModelForSequenceClassification. The texts were tokenized using AutoTokenizer, with padding and truncation set up for a max length of 512 for better GPU utilization. The classification head of the model was modified to 24 output neurons, each neuron having sigmoid activation for multilabel classification. The WeightedTrainer was applied to train, and it uses BCEWithLogitsLoss weighted with class weights to compensate for class imbalance. Input data were fed into the system using a custom PyTorch Dataset, converting texts to input IDs and attention masks, and labels into tensors. The Hugging Face Trainer took care of all optimization, validation, and checkpointing, while metrics such as Hamming loss and accuracy were calculated by thresholding the sigmoid outputs at 0.3 for binary predictions.

3.4.3 QARiB

The model was specifically designed for the Arabic language by researchers affiliated with the Qatar Computing Research Institute. In a manner akin to MARBERT, QARiB underwent pretraining on textual data encompassing both Modern Standard Arabic and a variety of Arabic dialects. The corpus of Modern Standard Arabic comprises news articles and subtitles from films and television, whereas the dialectal component consists of tweets. The architecture of this model mirrors that of Multilingual BERT. The most recent iteration of the model was pretrained on a dataset containing 14 billion tokens, amounting to approximately 127 gigabytes of text. This positions it as the second largest pretraining corpus, following MARBERT [27].

For this work, the fine-tuning of the QARiB model using the AutoModelForSequenceClassification class of the Hugging Face Transformers library was necessary. Input texts were tokenized using the associated AutoTokenizer, applying padding and truncation of up to 512 tokens, which is a good trade-off between model performance and computational efficiency on GPU-enabled platforms. The classification head comprised 24 output neurons, representing every target category, with a sigmoid activation to allow for multi-label classification. WeightedTrainer, a custom class, was used alongside the BCEWithLogitsLoss function with class weights

to counteract class imbalance. Input data were wrapped in a custom Dataset class in PyTorch that would tokenize text into input IDs and attention masks while converting label vectors into PyTorch tensors. The Hugging Face Trainer API was used to conduct the training, including gradient updates, validation, checkpointing, and metric computation, with Hamming loss and accuracy being calculated by thresholding sigmoid outputs at 0.3 to obtain binary predictions.

3.5 Training Strategy

The training strategy implemented for QARiB, MARBERT, and AraBERT in this code snippet is a generic multi-label text classification pipeline for Arabic texts. This strategy leverages the Hugging Face Transformers library and PyTorch to fine-tune pre-trained transformers on custom datasets with the following notable aspects :

3.5.1 Dataset Splitting Strategy

Using train-test-split from Scikit-learn, having set the random seed to 42 for reproducibility, the dataset was split into train (70%), validation (10%), and test (20%) sets.

3.5.2 Text Tokenization

The text inputs are tokenized using the AutoTokenizer from the Hugging Face Transformers library, with each model having its specific tokenizer (qarib/bert-base-qarib for QARiB, UBC-NLP/MARBERT for MARBERT, and aubmindlab/bert-base-arabertv02 for AraBERT). Tokenization includes padding and truncation to an upper limit of 512 tokens to maintain a nice trade-off between speed and performance.

3.5.3 Training Configuration

- Number of Epochs :5 epochs for training .
- Batch Sizes : The batch size is 32 for training
- Learning Rate : It is set at 3e-5
- Warmup Ratio : 0.1
- Weight Decay : It is set to 0.2 for regularization and prevents over-fitting.
- Mixed Precision Training : fp16=True accelerates training on GPU and saves memory.

3.5.4 Evaluation Metrics

A custom ‘compute-metrics‘ function calculates :

- Hamming loss
- Accuracy

3.6 Conclusion

This chapter involved a multi-label classification setup for Arabic language texts, wherein AraBERT, MARBERT, and QARiB models were fine-tuned on the NADiA1 dataset. With preprocessing, class-weighted training, and a set of optimized hyperparameters, the system can efficiently deal with complicated Arabic text classification. Supported by Pandas, NumPy, Scikit-learn, PyTorch, Transformers, and executed through the Kaggle platform, this guarantees excellent performance. The next chapter, i.e., Chapter 4, will delve into the efficacy of the system through experimental results.

CHAPTER 4

EXPERIMENTAL RESULTS

4.1 Introduction

This chapter has detailed the experimental evaluation carried out on a set of systems performing multi-label text classification for Arabic news, using AraBERT, MARBERT, and QARiB as transformer-based systems. It seeks to rate their performances by conducting an extensive study into the training, validation, and test metrics of the NADiA1 dataset; also, by directly evaluating sample predictions. The chapter thus builds a clear framework with structured tables and aggregate metrics within which the models derive their relative merits as solutions to the challenges of multi-label Arabic text classification.

4.2 Evaluation Metrics

Since multi-label classification involves awarding several category labels to each input, the performance evaluation of such models requires specific metrics which reflect those multi-dimensionalities. For an exhaustive evaluation, some of the most standard measures have been used, each describing a dissimilar property of the measure, namely the ones mentioned below :

Hamming Loss : Hamming loss is arguably the most popular metric for evaluating performance in Multi-label classification.This is expected, since it is simple to compute. The operator yields the symmetric difference between Yi, the actual label set of the ith instance, and Zi, the predicted label set. The |r| operator measures how many 1s are in this difference, which corresponds to the number of incorrect predictions. The total mistakes across the n instances are summed up and then normalized based on the number of labels and instances [28].

- N : The number of data instances in the multi-label dataset
- k : the total number of labels.
- Yi : The real labelset of the ith instance.
- Zi : The predicted labelset of the ith instance.

$$\textbf{HammingLoss} = \frac{1}{n \cdot k} \sum_{i=1}^{n} |Y_i \, \Delta \, Z_i| \tag{4.1}$$

Accuracy : In the multi-label field, Accuracy is defined as the proportion between the number of correctly predicted labels and the total number of active labels, in both real label set and the predicted one. The measure is computed by each instance and then averaged, as all example-based metrics [28].

$$\textbf{Accuracy} = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i \cap Z_i|}{|Y_i \cup Z_i|} \tag{4.2}$$

Precision and Recall : Precision is considered one of the more intuitive metrics to assess multi-label predictive performance. It is calculated as the proportion between the number of labels correctly predicted and the total number of predicted labels. Thus, it can be interpreted as the percentage of predicted labels that are truly relevant for the instance. This metric is usually used in conjunction with Recall which returns the percentage of labels correctly predicted among all truly relevant labels. That is, the ratio of true labels is given as output by the classifier [28].

$$\textbf{Precision} = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i \cap Z_i|}{|Z_i|} \tag{4.3}$$

$$\textbf{Recall} = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i \cap Z_i|}{|Y_i|} \tag{4.4}$$

F1 Score : F1 score, also known as balanced F-score or F1-measure The F1 score can be interpreted as a weighted average of the precision and recall, This way a weighted measure of how many relevant labels are predicted and how many of the predicted labels are relevant is obtained. An F1 score reaches its best value at 1 and worst score at 0. The relative contribution of precision and recall to the F1 score are equal [28].

$$\textbf{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.5}$$

4.3 Results and Analysis

The performance of this multi-label classification system was examined by running one of the transformer-based models (AraBERT, MARBERT, or QARiB) on the NADiA1 dataset, which contains 35,404 Arabic text files across 24 categories. The whole analysis is based on training, validation, and test metrics, with some sample predictions in order to determine how well the model can perform complex multi-label Arabic text classification. The results are presented in tabular form for further clarity, with a detailed interpretation of performance, strengths, and weaknesses.

4.3.1 AraBERT

4.3.1.1 Training and Validation Performance

Training lasted for 5 epochs, and metrics for training loss, validation loss, Hamming loss, and accuracy were recorded. These metrics are presented in Table 4.1, showing consistent improvement with each epoch.

Tab. 4.1 : Training and validation metrics over epochs.

Epoch	Training Loss	Validation Loss	Hamming Loss	Accuracy	Epoch Time (s)
1	0.1500	0.1156	0.0433	0.9567	1289.196
2	0.0995	0.0967	0.0396	0.9604	1289.196
3	0.0843	0.0909	0.0385	0.9615	1289.196
4	0.0748	0.0885	0.0383	0.9617	1289.196
5	0.0687	0.0884	0.0377	0.9623	1289.196

Training performances show steady loss decrease over epochs-from 0.1500 to 0.0687-limiting efficient parameter optimization. The validation losses had a similar downhill trend, ranging from 0.1156 to 0.0884, while they stabilized in the last two epochs, indicating that the model converged. Hamming loss dropped, more so from 0.0433 to 0.0377, implying less misclassification of labels and higher classification accuracy going from 0.9567 to 0.9623. The runtime consistency of around 1289.196 seconds per epoch for 2833 validation samples manifests computational efficiency.

4.3.1.2 Test Set Performance

Table 4.2 summarizes the evaluation over the test set with a Hamming loss of 0.0424 and an accuracy of 0.9576, demonstrating good generalization performance.

Tab. 4.2 : Test set performance metrics.

Metric	Value
Hamming Loss	0.0424
Accuracy	0.9576

Table 4.3 shows the classification report per-category, and includes precision, recall, and F1-score together with aggregated metrics for the test set (support : 19,132 labels).

Tab. 4.3 : Classification report for AraBERT test set performance.

Category	Precision	Recall	F1-Score	Support
News	0.70	0.92	0.80	3566
North Africa	0.89	0.88	0.89	1105
Levant	0.91	0.88	0.89	1134
Middle East	0.92	0.93	0.92	2995
The Americas	0.79	0.53	0.63	816
Research	0.44	0.71	0.54	161
Finance & Economy	0.75	0.64	0.69	519
War & Terrorism	0.73	0.66	0.70	1100
Gulf	0.90	0.84	0.87	877
Europe	0.82	0.50	0.63	1062
Political Figures	0.64	0.44	0.52	545
Iran	0.93	0.74	0.82	244
Technology	0.86	0.85	0.86	287
Russia	0.77	0.47	0.58	262
Sports	0.94	0.99	0.96	1635
Tennis	0.96	0.97	0.97	378
Football	0.84	0.94	0.89	932
English League	0.66	0.92	0.77	182
Arabian Sports	0.67	0.82	0.74	154
Spanish League	0.82	0.90	0.86	310
Health	0.52	0.67	0.58	146
East Asia	0.91	0.53	0.67	351
Environment	0.38	0.64	0.47	59
Other Countries	0.90	0.21	0.34	312
Micro Avg	0.81	0.81	0.81	19132
Macro Avg	0.78	0.73	0.73	19132

An average micro F1-score of 0.81 and a weighted average F1-score of 0.80 shows that the methods are balanced in their performances, whereas the macro-average of 0.73 points towards variability in those categories with low support. In the high prevalence categories, ”Sports” (F1: 0.96), ”Middle East” (F1: 0.92), and ”Tennis” (F1: 0.97) perform well due to high recall. Minority classes, on the contrary, see worse performances, with ”Environment” at a 0.47 F1 and ”Other Countries” at 0.34 F1, with particularly poor recall for ”Other Countries” (0.21), which is an indication of challenges in appropriately making predictions for underrepresented labels .

4.3.1.3 Sample Predictions Analysis

Figure 4.1 presents sample predictions, providing a side-by-side comparison of true and predicted labels for four test texts .

```
Predictions:

TEXT 1:
Text: ونزل الدولار اقل مستوي الاقتراع البريطاني بالموافقه الخروج عضويه الاتحاد الاوروب...
True: ['East Asia']
Pred: ['Finance & Economy']

TEXT 2:
Text: وسلط التقرير الضوء قصه جعفر تويكايا امضي يزيد حياته بخدم البحريه التركيه ينتقل ا...
True: ['News', 'Europe', 'Political Figures']
Pred: ['News', 'Europe']

TEXT 3:
Text: واضافت حركه النهضه وحزب المؤتمر الجمهوريه وحزب التكتل العمل والحريات بيان مشترك ...
True: ['News', 'North Africa']
Pred: ['News', 'North Africa']

TEXT 4:
Text: الوكيل ديميتري سيلوك السابع لاعب منتخب كوت ديفوار سيرحل بنسبه ابلغ سكاي سبورتس ا...
True: ['Sports', 'Football', 'English League']
Pred: ['Sports', 'Football', 'English League']
```

Fig. 4.1 : Sample predictions with true and predicted labels.

4.3.2 MARBERT

4.3.2.1 Training and Validation Performance

MARBERT was trained five epochs, and metrics such as training loss, validation loss, Hamming loss, and accuracy were recorded. Table 4.4 gives a summary of the performances, showing improvement through epochs.

Tab. 4.4 : Training and validation metrics for MARBERT across five epochs

Epoch	Training Loss	Validation Loss	Hamming Loss	Accuracy	Epoch Time (s)
1	0.1788	0.1441	0.0512	0.9488	1351.14
2	0.1188	0.1144	0.0446	0.9554	1351.14
3	0.0948	0.1073	0.0432	0.9568	1351.14
4	0.0790	0.1036	0.0419	0.9581	1351.14
5	0.0688	0.1051	0.0421	0.9579	1351.14

The decrease in training loss was from 0.1788 to 0.0688, from which it can be understood that the parameters were well optimized. The validation loss decreased from 0.1441 to 0.1051, albeit with a slight increase in the final epoch that may suggest overfitting or noise in the validation set. Hamming loss improved from 0.0512 to 0.0419, meaning the mislabeled instances have reduced ; meanwhile, the accuracy has gone up from 0.9488 to 0.9579. About 1351.14 second per epoch for 2833 validation samples demonstrates computational efficiency.

4.3.2.2 Test Set Performance

The summary of MARBERT performance on the test set is given in Table 4.5, where it recorded a Hamming loss of 0.0501 and accuracy of 0.9499.

Tab. 4.5 : Test set performance metrics for MARBERT.

Metric	Value
Hamming Loss	0.0501
Accuracy	0.9499

The classification report in Table 4.6 presents the details of the per-category precision, recall, and F1-score, along with aggregate metrics for the test set (support : 19,132 labels).

Tab. 4.6 : Classification report for MARBERT test set performance.

Category	Precision	Recall	F1-Score	Support
News	0.70	0.92	0.79	3566
North Africa	0.87	0.90	0.88	1105
Levant	0.85	0.90	0.88	1134
Middle East	0.87	0.94	0.90	2995
The Americas	0.81	0.29	0.43	816
Research	0.80	0.28	0.41	161
Finance & Economy	0.83	0.29	0.43	519
War & Terrorism	0.65	0.55	0.59	1100
Gulf	0.91	0.77	0.84	877
Europe	0.75	0.35	0.47	1062
Political Figures	0.86	0.06	0.11	545
Iran	0.94	0.60	0.73	244
Technology	0.86	0.84	0.85	287
Russia	0.89	0.06	0.12	262
Sports	0.93	0.98	0.96	1635
Tennis	0.91	0.96	0.93	378
Football	0.78	0.97	0.86	932
English League	0.68	0.85	0.75	182
Arabian Sports	0.74	0.69	0.72	154
Spanish League	0.67	0.93	0.78	310
Health	0.57	0.56	0.56	146
East Asia	0.99	0.19	0.32	351
Environment	0.51	0.47	0.49	59
Other Countries	0.89	0.03	0.05	312
Micro Avg	0.80	0.74	0.77	19132

4.3.2.3 Sample Predictions Analysis

Figure 4.2 summarizes sample predictions by comparing true and predicted labels for three test texts.

```
Sample Predictions:

Sample 1:
Text: ونزل الدولار اقل مستوي الاقتراع البريطاني بالموافقه الخروج عضويه الاتحاد الاوروب...
True: ['East Asia']
Pred: ['Finance & Economy']

Sample 2:
Text: وسلط التقرير الضوء قصه جعفر تويكايا امضي يزيد حياته يخدم البحريه التركيه ينتقل ا...
True: ['News', 'Europe', 'Political Figures']
Pred: ['News', 'Europe']

Sample 3:
Text: واضافت حركه النهضه وحزب المؤتمر الجمهوريه وحزب التكتل العمل والحريات بيان مشترك ...
True: ['News', 'North Africa']
Pred: ['News', 'North Africa', 'Middle East']
```

Fig. 4.2 : Sample predictions with true and predicted labels for MARBERT.

4.3.3 QARiB

4.3.3.1 Training and Validation Performance

Training the model for five epochs involved measuring training loss, validation loss, Hamming loss, and accuracy. Table 4.7 summarizes the performance, showing consistent improvement.

Tab. 4.7 : Training and validation metrics for QARiB across five epochs

Epoch	Training Loss	Validation Loss	Hamming Loss	Accuracy	Epoch Time (s)
1	0.1568	0.1256	0.0470	0.9530	1417.896
2	0.1051	0.1035	0.0429	0.9571	1417.896
3	0.0864	0.0972	0.0407	0.9593	1417.896
4	0.0740	0.0967	0.0408	0.9592	1417.896
5	0.0658	0.0976	0.0408	0.9592	1417.896

Training loss went from 0.1568 to 0.0658 as the parameters were better optimized; validation loss went down from 0.1256 to 0.0967 but slightly rose back to 0.0976 in the last epoch, hence hinting at mild overfitting or variation in the validation set. Hamming loss got reduced from 0.0470 to 0.0408 with accuracy improving from 0.9530 to 0.9592, together indicating the classification of more labels correctly as time passed, while the run-time of 1417 seconds with 2833 samples for validation is a bit slower than others. This might have been a result of QARiB's architecture .

4.3.3.2 Test Set Performance

Table 4.8 summarizes QARiB's test performance wherein it scored a Hamming Loss of 0.0452 and an accuracy of 0.9548, which implies the model generalizes well on unseen data.

Tab. 4.8 : Test set performance metrics for QARiB.

Metric	**Value**
Hamming Loss	0.0452
Accuracy	0.9548

The classification report of precision, recall, and F1-score for each category and aggregated metrics for the entire test set is presented in Table 4.9 (support : 19,132 labels).

Tab. 4.9 : Classification report for QARiB test set performance.

Category	Precision	Recall	F1-Score	Support
News	0.68	0.94	0.79	3566
North Africa	0.88	0.89	0.89	1105
Levant	0.90	0.89	0.89	1134
Middle East	0.91	0.93	0.92	2995
The Americas	0.83	0.39	0.54	816
Research	0.38	0.70	0.49	161
Finance & Economy	0.77	0.57	0.65	519
War & Terrorism	0.69	0.65	0.67	1100
Gulf	0.88	0.85	0.87	877
Europe	0.72	0.61	0.66	1062
Political Figures	0.78	0.22	0.35	545
Iran	0.91	0.75	0.82	244
Technology	0.81	0.87	0.84	287
Russia	0.81	0.39	0.53	262
Sports	0.95	0.98	0.96	1635
Tennis	0.94	0.97	0.95	378
Football	0.84	0.93	0.88	932
English League	0.68	0.84	0.75	182
Arabian Sports	0.66	0.76	0.71	154
Spanish League	0.70	0.95	0.80	310
Health	0.44	0.82	0.58	146
East Asia	0.91	0.62	0.74	351
Environment	0.45	0.47	0.46	59
Other Countries	0.91	0.28	0.42	312
Micro Avg	0.79	0.81	0.80	19132
Macro Avg	0.77	0.72	0.72	19132

4.3.3.3 Sample Predictions Analysis

Figure 4.4 summarizes the sample predictions by comparing the true labels to the predicted labels for three test texts, providing some qualitative insight into QARiB's performance.

```
Sample Predictions:

Sample 1:
Text: ونزل الدولار اقل مستوي الاقتراع البريطاني بالموافقه الخروج عضويه الاتحاد الاوروب...
True: ['East Asia']
Pred: ['Finance & Economy']

Sample 2:
Text: وسلط التقرير الضوء قصه جعفر تويكايا امضي يزيد حياته يخدم البحريه التركيه ينتقل ا...
True: ['News', 'Europe', 'Political Figures']
Pred: ['News', 'Europe']

Sample 3:
Text: واضافت حركه النهضه وحزب المؤتمر الجمهوريه وحزب التكتل العمل والحريات بيان مشترك ...
True: ['News', 'North Africa']
Pred: ['News', 'North Africa']
```

Fig. 4.3 : Sample predictions with true and predicted labels for QARiB..

4.4 Comparative Analysis of AraBERT, MARBERT, and QARiB

This section compares the performances of the three transformer-based models of AraBERT , MARBERT , and QARiBacross the 35,404 Arabic text files spanning 24 categories . The comparison checks their training, validation, and test metrics, classification performance, and sample predictions to view where each model stands in strengths and weaknesses in multi-label Arabic text classification. Key metrics are summarized in tables and discussed in detail in the subsequent paragraphs.

4.4.1 Training and Validation Performance

All models were trained for five epochs, during which training loss, validation loss, Hamming loss, and accuracy were noted. Table 4.10 shows the comparative performance of each model during the last epoch (Epoch 5).

Tab. 4.10 : Final epoch (Epoch 5) training and validation metrics for AraBERT, MARBERT, and QARiB.

Epoch	Training Loss	Validation Loss	Hamming Loss	Accuracy	Runtime
AraBERT	0.0687	0.0884	0.0377	**0.9623**	6445.98s
MARBERT	0.0688	0.1051	0.0421	0.9579	6755.72s
QARiB	0.0658	0.0976	0.0408	0.9592	7089.48s

AraBERT acquires the lowest validation loss (0.0884) and Hamming loss (0.0377), with the highest accuracy (0.9623), suggesting better convergence and generalization on the validation set. MARBERT shares a similar training loss (0.0688) but has a higher validation loss (0.1051) and Hamming loss (0.0421) and lower accuracy (0.9579), thus indicating worse generalization. QARiB has the lowest training loss (0.0658) but has a higher validation loss (0.0976) than AraBERT and Hamming loss (0.0408) and accuracy (0.9592) between the other two Longer runtime (7089.48s) point to a higher computational cost, possibly due to its architecture or complexity in the pre-training corpus.

4.4.2 Test Set Performance

Table 4.11 compares test set performances emphasizing Hamming loss, accuracy, and aggregated F1-scores extracted from the classification reports.

Tab. 4.11 : Final test set performance comparison between AraBERT, MARBERT, and QARiB

Model	Hamming Loss	Accuracy	Micro F1	Macro F1	Weighted F1	Samples F1
AraBERT	0.0424	0.9576	0.81	0.73	0.80	0.78
MARBERT	0.0501	0.9499	0.77	0.62	0.73	0.73
QARiB	0.0452	0.9548	0.80	0.72	0.79	0.77

With the least Hamming loss (0.0424) and the best accuracy (0.9576), AraBERT applies a balanced performance across labels with a 0.81 micro F1-score. QARiB is a close second with a Hamming loss of 0.0452, an accuracy of 0.9548, and a micro F1-score of 0.80. MARBERT underperforms with a higher Hamming loss (0.0501), lower accuracy (0.9499), and a micro F1-score of 0.77, implying inferior performances overall. Macro F1-scores manifest disparities; MARBERT's 0.62 is substantially lower than AraBERT's 0.73 and QARiB's 0.72, suggesting that MARBERT has issues with the low-support classes.

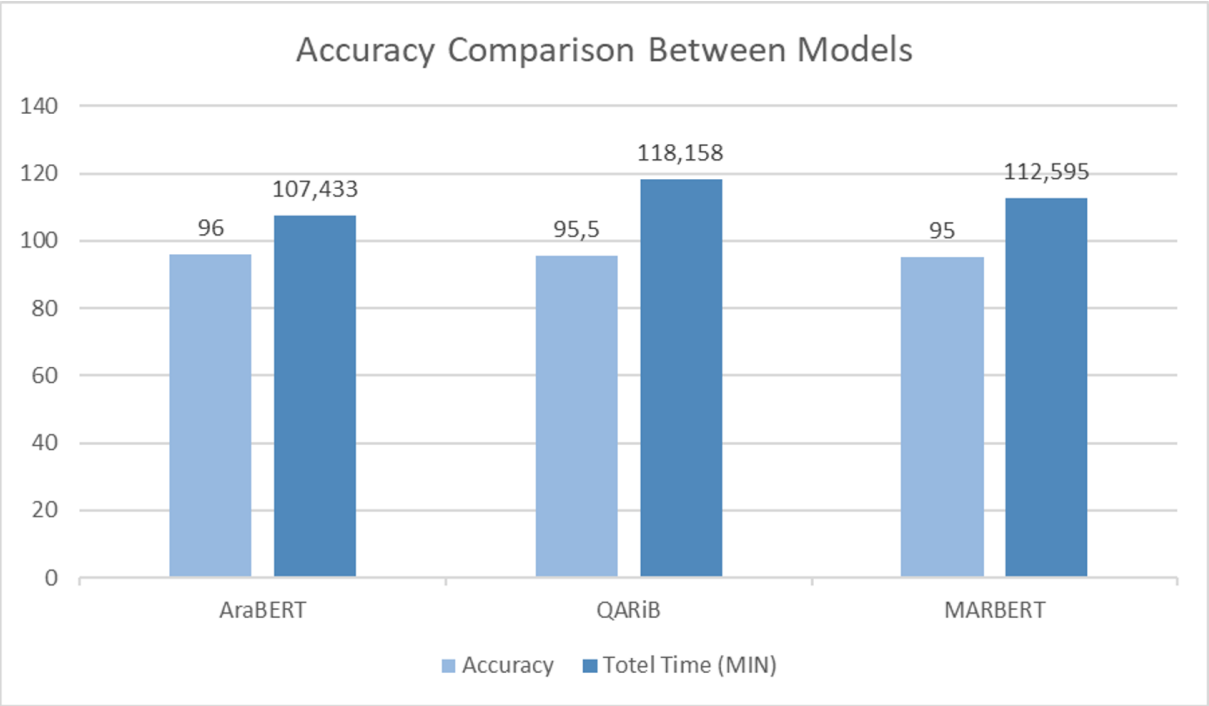

Fig. 4.4 : Accuracy and training time comparison Between Models .

4.5 Conclusion

In the experimental results, it is seen that AraBERT, MARBERT, and QARiB can fairly well accomplish multi-label Arabic text classification. The performance of AraBERT comes on top, followed closely by QARiB, and MARBERT exhibiting some disadvantages for low-support categories. The models perform well in categories having high prevalence such as ”Sports” and ”Middle East” due to high recall, but poorly on minority classes such as ”Environment” and ”Other Countries,” even with class weighting. The results show the strengths of transformer-based models for Arabic text and call for further research on additional techniques such as adaptive thresholding or data augmentation to improve performance on underrepresented labels.

GENERAL CONCLUSION

This study tried to address the issues of multi-label Arabic text categorization deep learning methods, especially the transformers AraBERT, MARBERT, and QARiB applied on a huge dataset, NADiA1, consisting of 35,404 Arabic text files distributed into 24 main categories. This study aimed to build and test a system for associating Arabic texts with multiple suitable categories, The experimental analysis revealed that all 3 models are capable of handling multi-label

classification; however, AraBERT scored the highest with 95.76 % accuracy and a micro F1 score of 0.81, while QARiB with accuracy at 95.48% and micro F1 at 0.80 closely followed. MARBERT, meanwhile, worked at almost the same level with an accuracy of 94.99% and a micro F1 score of 0.77 but was hamstrung by low support categories. Categories with high prevalence like Sports (F1: 0.96 in all models) and Middle East (F1: 0.90-0.92) were fairly good performers with high recall, whereas minority category classes such as Environment (F1: 0.46-0.49) and Other Countries (F1: 0.05-0.42) bore the brunt of shortcomings, with MARBERT faring the worst owing to extremely low recall (e.g., 0.03 for Other Countries). Sample predictions revealed shared sets of challenges on the detection of nuanced or underrepresented labels such as misclass.

Thus, this study offers a useful and well-performing method for multi-label classification for Arabic texts. In so doing, it establishes that pre-trained transformer models prove very useful in Arabic NLP and shows that threshold tuning is an effective technique for offsetting performance issues caused by imbalanced classes.

The main challenges encountered throughout this work included imbalanced class distributions, high computational demands, and limited availability of well-annotated Arabic datasets. Further, adjusting the classification thresholds for each label was found to be a complicated process that was also time-intensive.

In future research, our forced expansion will encompass real-time classification of Arabic content, exploration of more complex transformer architectures, and eventual creation of an interactive platform for multi-label classification tasks. Further work can also be directed toward integrating explainability methods and expanding the dataset with more diverse sample sets of Arabic texts across different dialects and terminologies.

REFERENCES

[1] K. Randhe, Y. Gade, A. Chhatre, and A. Sahani, "Natural language processing," *International Journal of Research Publication and Reviews*, vol. 4, no. 6, pp. 2034–2045, June 2023. [Online]. Available : http://www.ijrpr.com

[2] Y. Kang, Z. Cai, C.-W. Tan, Q. Huang, and H. Liu, "Natural language processing (NLP) in management research : A literature review," *Journal of Management Analytics*, vol. 7, no. 2, pp. 139–172, Apr. 2020, 2025-05-02. [Online]. Available : https://www.tandfonline.com/doi/full/10.1080/23270012.2020.1756939

[3] M. S. H. Ameur, "Exploring deep learning techniques in the task of English-Arabic Machine Translation," phdthesis, Université des Sciences et de la Technologie Houari Boumediene (Algérie), Jun. 2020, 2025-05-04. [Online]. Available : https://theses.hal.science/tel-03202829

[4] K. Ayoub and S. Ramzi, "Deep learning models applied to arabic quranic text," Master's thesis, UNIVERSITY OF MOHAMED BOUDIAF – MSILA, 2024.

[5] B. MAISSA, "Implémentation d'une méthode hybride (morphologique statistique) pour l'analyse des mots arabes," Master's thesis, UNIVERSITY OF MOHAMED BOUDIAF – MSILA, 2017.

[6] Z. Seloua and Z. Sirine, "Arabic poems classification according to their eras and topics," Master's thesis, Université de Blida 1–Saad Dahlab, 2020.

[7] S. Kamruzzaman, "Text classification using artificial intelligence," *arXiv preprint arXiv :1009.4964*, 2010.

[8] P. D. Moral, S. Nowaczyk, and S. Pashami, "Why Is Multiclass Classification Hard ?" *IEEE Access*, vol. 10, pp. 80 448–80 462, 2022. [Online]. Available : https://ieeexplore.ieee.org/document/9839332/

[9] H. Alfigi, "MULTI-LABEL AND SINGLE-LABEL TEXT CLASSIFICATION USING STANDARD MACHINE LEARNING ALGORITHMS AND PRE-TRAINED BERT TRANSFORMER."

[10] Ai machine learning presentation diagrams. [Online]. Available :https://www.infodiagram.com/diagrams/ai-diagrams-machine-learning-ppt-template/. InfoDiagram.com. Accessed on : Apr. 9, 2025.

[11] B. alsukhni, "Multi-label arabic text classification based on deep learning," in *2021 12th International Conference on Information and Communication Systems (ICICS)*. IEEE, 2021, pp. 475–477.

[12] A. Al-Qarqaz and M. Abdullah, "Team r00 at mowjaz multi-topic labelling task for arabic articles," 05 2021, pp. 462–465.

[13] H. El Rifai, L. Al Qadi, and A. Elnagar, "Arabic text classification : the need for multi-labeling systems," *Neural Computing and Applications*, vol. 34, no. 2, pp. 1135–1159, 2022.

[14] A. S. Abid and Z. C. B. Othmane, "Comparing deep learning models for multi-label classification of arabic abusive texts in social media." in *ICSOFT*, 2022, pp. 374–381.

[15] M. El Abdi, B. Smine, S. B. Yahia, and H. K. B. Ayed, "Hana-mc : Heading of arabic news analysis by multi-label classification," *Procedia Computer Science*, vol. 246, pp. 3556–3565, 2024.

[16] H. Rahman and S. Baawi, "A proposed arabic text classification model using multi-label system," *Journal of Al-Qadisiyah for Computer Science and Mathematics*, vol. 15, 09 2023.

[17] N. Aljedani, R. Alotaibi, and M. Taileb, "Hmatc : Hierarchical multi-label arabic text classification model using machine learning," *Egyptian Informatics Journal*, vol. 22, no. 3, pp. 225–237, 2021.

[18] "What is python ? executive summary," [Online].Available :https://www.python.org/doc/essays/blurb/, accessed on :2025-04-27.

[19] "Package overview — pandas 2.2.3 documentation," [Online].Available :https://pandas.pydata.org/docs/getting_started/overview.html, accessed on :2025-04-28.

[20] "What is NumPy ? — NumPy v2.2 Manual," [Online].Available :https://numpy.org/doc/stable/user/whatisnumpy.html, accessed on :2025-04-28.

[21] "Getting Started," [Online].Available :https://scikit-learn/stable/getting_started.html, accessed on :2025-04-28.

[22] "PyTorch," [Online].Available :https://ai.meta.com/tools/pytorch, accessed on :2025-06-09.

[23] “Transformers,” [Online].Available :https://huggingface.co/docs/transformers/en/index, accessed on :2025-04-30.

[24] “What is a kaggle? |kaggle,” [Online].Available :https://www.kaggle.com/general/328265, accessed on :2025-04-27.

[25] “Learn LaTeX in 30 minutes,” [Online].Available :https://www.overleaf.com/learn/latex/Learn_LaTeX_in_30_minutes, accessed on :2025-04-27.

[26] R. Al-Debsi, A. Elnagar, and O. Einea, “NADiA : News Articles Dataset in Arabic for Multi-Label Text Categorization,” vol. 2, Sep. 2019, publisher : Mendeley Data. [Online]. Available : https://data.mendeley.com/datasets/hhrb7phdyx/2

[27] A. S. Alammary, “BERT Models for Arabic Text Classification : A Systematic Review,” *Applied Sciences*, vol. 12, no. 11, p. 5720, Jan. 2022, number : 11 Publisher : Multidisciplinary Digital Publishing Institute. [Online]. Available : https://www.mdpi.com/2076-3417/12/11/5720

[28] M. I. AMROUCHE and A. D. CHOUTRI, “Multi-label classification of social networks’ users based on services,” Master’s thesis, UNIVERSITY YAHIA FARES OF MEDEA Faculty of Science, 2 0 1 9 - 2 0 2 0.

ABSTRACT

In multi-label text classification, multiple related labels are assigned to relevant documents for more refined categorization. The research attempts to build a system that can effectively categorize Arabic texts into multiple themes using the multi-label dataset NADiA1, which contains 35,404 files across 24 categories. Deep learning approaches, encompassing QARiB, MARBERT, and AraBERT, were used, with data preprocessing conducted using the pyArabic package to maintain text quality. The models were assessed by accuracy, precision, recall, and Hamming loss. Among these, transformer-based AraBERT outclassed its peers by giving 95.76% accuracy and a micro F1-score of 0.81, followed by QARiB (95.48% accuracy and 0.80 micro F1-score) and MARBERT (94.99% accuracy and 0.77 micro F1-score). This study lays emphasis that deep transformer-based learning techniques are highly effective in multi-label Arabic text classification, with AraBERT showing the ability to better handle linguistic complexities.

Keywords : Multi-label Text Classification, Deep Learning, QARiB, MARBERT, AraBERT, Arabic NLP, Arabic text, transformers.

ملخص

في تصنيف النصوص متعدد التسميات، يتم تخصيص عدة تسميات ذات صلة لمستند واحد، مما يتيح تصنيفًا دقيقًا ومفصلاً. تهدف هذه الدراسة إلى تطوير نظام فعال لتصنيف النصوص العربية إلى فئات موضوعية متعددة باستخدام قاعدة بيانات NADiA1، التي تضم 35,404 ملفات موزعة عبر 24 فئة في صيغة متعددة التسميات. تم تطبيق أساليب التعلم العميق، بما في ذلك QARiB وMARBERT وAraBERT، مع معالجة مسبقة للبيانات باستخدام حزمة pyArabic لضمان جودة النصوص. تم تقييم أداء النماذج من خلال مقاييس مثل الدقة، والتدقيق، والاستدعاء، وخسارة هامينغ. أظهرت النتائج أن نموذج AraBERT القائم على المحولات تفوق على غيره، محققًا دقة 95.76% ودرجة F1 ميكرو 0.81، تلاه QARiB بدقة 95.48% ودرجة F1 ميكرو 0.80، ثم MARBERT بدقة 94.99% ودرجة F1 ميكرو 0.77. تؤكد هذه الدراسة فعالية تقنيات التعلم العميق القائمة على المحولات في تصنيف النصوص العربية متعددة التسميات، مع تفوق AraBERT في التعامل مع التعقيدات اللغوية.

الكلمات المفتاحية : تصنيف النصوص متعدد التسميات، التعلم العميق، QARiB، MARBERT، AraBERT معالجة اللغة الطبيعية العربية، النصوص العربية، المحولات.