Available online at www.qu.edu.iq/journalcm
JOURNAL OF AL-QADISIYAH FOR COMPUTER SCIENCE AND MATHEMATICS
ISSN:2521-3504(online) ISSN:2074-0204(print)

A Proposed Arabic Text Classification Model using Multi-Label System

Hussain A.Rahman[a]*,Salwa S. Baawi

[a]*College of computer Science and Information Technology, University of AL-Qadisiyah.Email:com21.post2@qu.edu.iq*

[b]*College of computer Science and Information Technology, University of AL-Qadisiyah.Email:salwa.baawi@qu.edu.iq*

ARTICLE INFO

Article history:
Received: 20 /06/2023
Rrevised form: 11 /08/2023
Accepted : 15 /08/2023
Available online: 30 /09/2023

Keywords:

Text classification

Arabic text classification

Single-label text classification

Multi-label text classification

Feature selection

ABSTRACT

Multi-label text classification has grown in popularity in recent years, with each document being assigned numerous categories simultaneously. The Arabic Language has a very complex morphology and a vibrant nature; nonetheless, there needs to be more research on this topic for the Arabic Language. As a result, this study aims to present a method for the multi-label classification of Arabic texts based on binary relevance and the label power set transformation method. Three classification classifiers: namely logistics regression(LR), Random forest (RF), and multinomial naïve Bays (MNB), were experimentally assessed in this thesis. Furthermore, chi-square feature selection was investigated to improve the performance of the proposed model. The experimental results are implemented in Python programming using the "RTANews" multi-label Arabic text classification dataset. The results suggest that binary relevance combined with logistics regression produces the best results. It performed well, with an averaged micro-Recall of 0.8646. At the same time, the best result was produced by label power-set with the same algorithm and metrics of 0.8418 for the suggested multi-label Arabic text classification model.

MSC..

https://doi.org/10.29304/jqcm.2023.15.3.1269

1. Introduction

Text classification, also called Text categorization, is one of the most critical applications of Natural language processing used for handling large volumes of text documents available recently due to the rapid growth of the internet, social media, and other technologies today. Text classification is a process of classifying documents into one or more categories based on the document content itself[1]. this gives rise to many applications of text classification such as sentiment analysis[2], spam filtering[3] , and text summarization[4].

Text classification models undergo two stages: training (labeled data) and testing (new data). The training stage involves labeled data, while the testing stage evaluates the model using new data. The supervised machine learning approach requires labeled data, while unsupervised machine learning uses statistical techniques to detect

∗Corresponding author

Email addresses:

Communicated by ‘sub etitor’

similarities among documents without predefined labeled data. This approach saves human effort and is less time-consuming using automation than manual methods[5].

In text classification, the number of labels in each instance determined the distinction between single-label text classification and multi-label text classification. The single-label text classification, where the most suitable single category is assigned to each instance based on content or context, such as sentiment analysis, where each instance could include positive, natural, and negative[6]. The multi-label text classification single instance can be assigned to one or more than one simultaneously; for example, covid-19 articles may classify COVID-19 and economic categories simultaneously. Single-label text classification algorithms struggle with multi-label problems. Problem transformation techniques convert multi-label problems into single-label problems, enabling classical machine learning algorithms to adapt or take on multi-label classification problems[7].

The Arabic language is spoken by around 380 million people, making it one of the world's foremost and most widely, and considered the sixth formal language of the United Nations. Arabic is rich in morphology and orthography. The 28 letters in the alphabet are read and written from right to left. There were only two. Genders in it (feminine and masculine).Singular, dual, and Plural nouns in Arabic There are three syntactic classes in Arabic: nonnative, accusative, and genitive. No capital letters. It also uses diacritics to hint at minor vowel characters like"fatha, kasra, damma, sukun, shadda, and tanween[8].

This paper presented a proposed Arabic text classification model for a multi-label problem (ATCM) based on applying binary relevance (BR) to divide a multi-label classification problem into several single-label problem and Label Powerset (LP) converts a multi-label problem to a multi-class problem.

The following is the order of the paper: Section 2, the relevant related works; Section 3 methodology, conclusion, and future work are covered in Section 4.

2. Related Works

The researchers focused on the single-label classification in the Arabic Language [6] making it rich in studies. Meanwhile, the Arabic Language lacks sufficient studies for multi-label text classification, unlike other languages such as English[4], Portuguese[5], etc. Therefore, this thesis, focuses on multi-label classification which, is a sub-domain of text classification as a strategy to handle a real-life problem such as text classification when a document or text belongs to multiple labels simultaneously.

For instance, [9]the first study in Arabic proposed research on binary and multi-class classification transformation methods. They used MEKA software to transform the multi-label classification of the dataset of 10k news articles from the BBC Arabic website into single-label classification using label combination (LC), Binary Relevance (BR), and ranking and threshold (RT). The authors used Four classifiers: Support vector machine (SVM), Naïve Bayes (NB), K-nearest neighbor (KNN), and Decision Tree (DT), to classify the dataset into five classes. The results showed that combining the label power set with SVM achieved the highest ML- accuracy at 71%.

The authors of [10]utilized a previously used dataset in [9] and constructed a new multi-label classification based on binary relevance. They employed three classifiers including : Support vector machine (SVM) , Naïve Bayes (NB) and K-nearest neighbor (KNN) and examined the effect of feature selection using three methods (chi-square, odd ratio, and mutual information). The evaluation measures used were precision, recall, and f-measure. The study found that the best performance was achieved when a set of single-label classifiers were merged with the chi-square method, resulting in an f-measure of 86.8%.

In [11], the authors studied the multi-label classification of Arabic article articles. They used the KNN, Random Forest (RF), and Decision Tree (DT) classifiers to address the multi-label classification problem. They applied them to a dataset of 10,997 Arabic news articles collected from CNN Arabic divided into six classes. The results showed that the DT classifier outperformed the KNN and RF classifiers, with precision, recall, F1-score, and hamming loss values of38.8%, 35.4%, 37%, and 0.3%, respectively.

In [12], the authors proposed a novel method for classifying Arabic text using a lexicon-based system that employs multiple lexicons, including both stemming and un-stemming. The method works by matching the words in each lexicon with the words in the given data and then classifying the data based on term frequency (TF). Five labels were selected based on count values. The authors collected 4720 articles from the BBC Arabic website they used. Four evaluation measures (ML accuracy, exact match, hamming loss, execution time) to compare their method to a

corpus-based method. The experiments showed that the lexicon-based method outperformed the corpus-based method, with a 31% improvement in ML accuracy.

besides, the authors in [13], Introduced a benchmark dataset from the site "RT Arabic News." The dataset consistent with 23,837 Arabic articles falls under 40 categories. They investigated four transformation-based algorithms: binary classification approaches like Binary Relevance (BR), Classifier Chains (CC), and Calibrated Ranking by Pairwise Comparison (CRPC), compared with the adaption algorithm: Instance-Based Learning by Logistic Regression Multi-label (IBLRML) , and Binary Relevance kNN (BRkNN), and RFBoost. They used three classifiers RF, KNN, and SVM. The best Micro-averaged Precision score obtained by KNN was 90.10% in the adaptation-based algorithm and 85.66% in transformation-based algorithms.

Also[14], the author built a multi-label Arabic text classification model. The author used SVM, logistic regression, and multilayer perceptron classifiers to evaluate the dataset consisting of 9.590 articles distributed over ten classes. The result showed that the SVM classifier achieved the best performance, with 82.2%.

In[15], the authors presented a hierarchical multi-label text classification study to classify Arabic language text. They used the Islamic requests (FATWA) appropriate hierarchical structure by using the HOMER algorithm. They used Hamming loss, H-loss, ML accuracy, and subset accuracy) and micro-averaged recall, micro-averaged precision, and micro-averaged F-measure. The result showed that the Micro-averaged F-measure best performance was achieved with 0.8536%.

The lack of studies resulted in a lack of published studies regarding the classification of multi-label texts in the Arabic Language to the need for more available and published datasets for researchers except for RTANews dataset. Therefore, this paper utilizes the same dataset Al-Salemi et al. presented in 2019.

3. Methodology

This section explains the theoretical foundations of all the research methodologies and materials.

3.1 Arabic Text Classification model

This section presents the Arabic text classification model. This model is based on applying the binary relevance and label power set problem transformation technique in order to enable the classical machine learning algorithm to deal with multi-label problem. Because of these algorithms only used to single label classification. In order to overcome the problem of multi-label classification, this proposed ATCM model that aims to classify Arabic text documents into multiple labels based on contents of these documents. To improve multi-label model performance utilizing multiple classifiers, the ATCM model includes pre-processing steps and feature selection. This suggested model focuses on flat classification datasets in modern standard Arabic. It employs multi-label classification metrics that have been established for multi-label classification. Figure 3.1 depicts the proposed Arabic text classification (ATCM) architecture. It steps include dataset Arabic documents, text pre-processing, feature extraction, feature selection, multi-label classification, and performance evaluation.

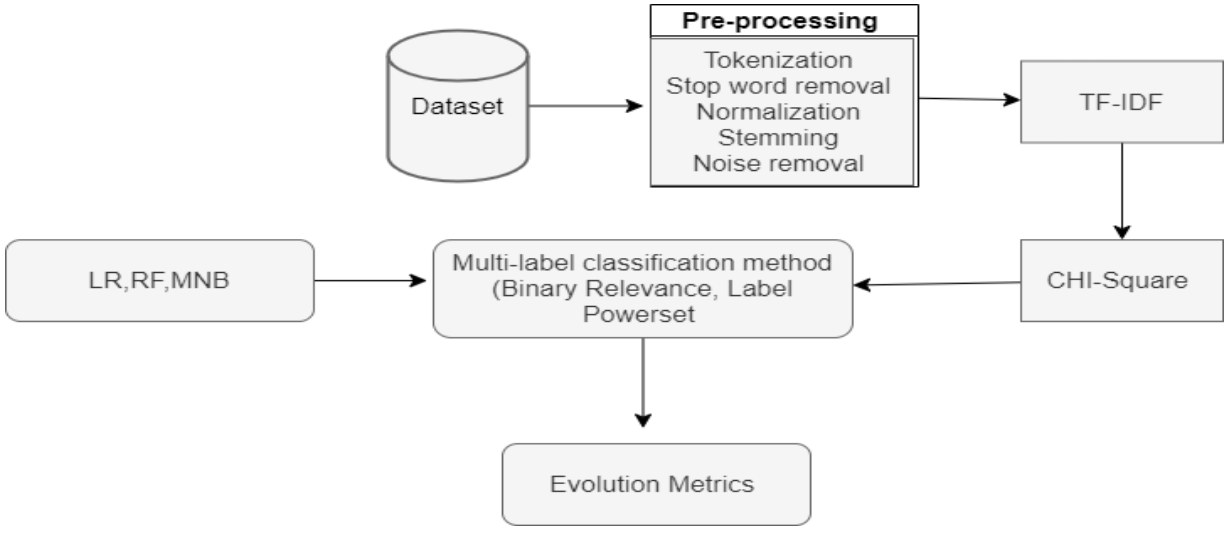

FIGURE 3.1 ATCM MODEL

3.2 **Dataset Description**

This section describes the dataset which consist of two parts: the first part focus on the training files named "training" and the other part combines 40 files named "test". We observed that there are 40 sub-categories in each part represented by text files (.txt).

The percentage of text files that fall under one sub-category is 85.48%, while the percentage of text files that fall under two categories is only 13.04%, and the number of text files that fall under categories 3, 4, and 5 is 329, 19, and 1, respectively. Figure 3.2 shows the percentage of the distribution of text files on the sub-categories of the data set, where the imbalance of the data is clearly shown, as some categories contain a greater number of other categories. In The proposed model, training set and test set files convert into CSV (Comma-Separated Values) format for easier handling and preparation for further steps then merged training and test file in single csv file then split them in two portion 70% for training and 30% for testing .

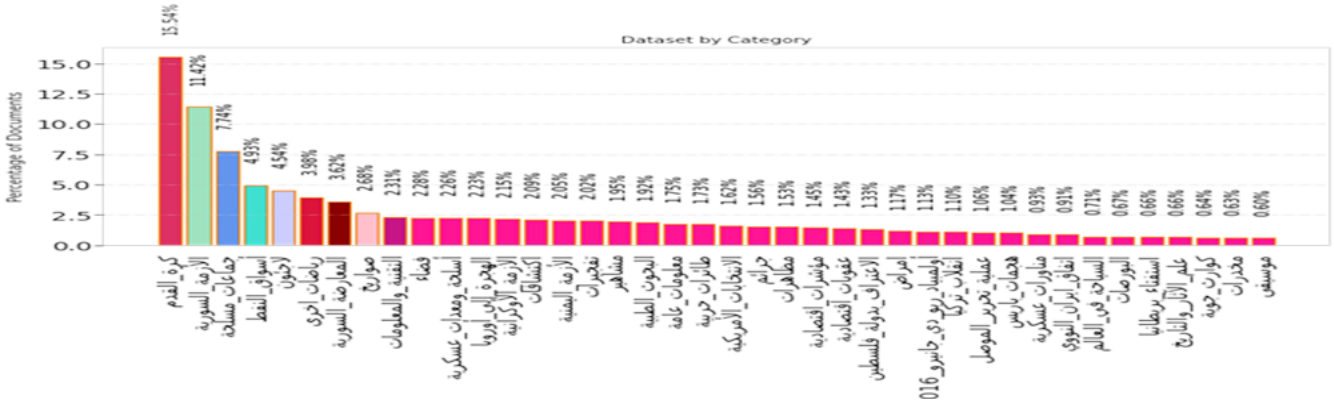

FIGURE 3.2 DISTRIBUTION OF DOCUMENTS PER CATEGORY IN RTANEWS

3.2 Text Preprocessing

This section explains one of the most critical phases in ATCM, which is pre-processing. The essential advantage of employing data preprocessing is that it minimizes the total number of features in the dataset while improving the performance of classifiers in terms of resource requirements and classification precision. Figure 3.3 depicts data pre-processing as five fundamental processes: tokenization, stop words removal, normalization, stemming, and noise removal to create cleaned text that is then employed in the classification phase.

FIGURE 3.3MAIN PROCESSES OF PREPROCESSING PHASE

a. Tokenization

The process identifies individual words or tokens using separators like white spaces and punctuation marks. The Arabic Language is highly derivational and inflectional, with various word forms and diacritics. The exact three-letter origin can result in different words with different meanings, and the same word may have multiple variations with different suffixes, affixes, and prefixes.

b. Stop word removal

Stop words are a preprocessing process in machine learning that removes frequent and insignificant terms from text. In Arabic, this process involves removing subject pronouns, objective pronouns, and question adjectives to improve computing performance and focus on basic patterns or attributes. The process involves removing subject pronouns(e.g. انتم ، نحن ، هي ، هو ،انت ،أنا), objective pronouns(e.g. لهم ،لكم ،لنا ،له ،لي ،لك), and question adjectives (e.g. من ،متى ،اين ،ما ,له ،لي ،لك) to improve the overall text.

c. Normalization

In this stage, the normalization is conducted by the transformation of the letter in the text into the canonical form or by removing the diacritics[67]., for example:

1. Alif letter different forms "آ" ," أ " ",أ" ","إ replaced with "ا".

2. Yah character " ئ" replaced with " ى".

3. Tah marbuta character "ة" replaced with "ه"

4. Strip The Tashkeel and Tatweel

d. Stemming

Stemming is the process of returning a word to its base root by mapping derived tokens to their base stem form. The Farasa light stemming algorithm is employed to reduce dimensionality by removing suffixes, prefixes, and affixes from words.

e. Noise Removal

Noise removal is the process of eliminating components that obstruct text analysis. ATMC uses regular expressions to clean text from noise, including number removal, non-Arabic letter removal, and punctuation and special character removal. This involves removing Arabic and English digits, non-Arabic letters, and special characters like $, %, @, and &.

3.3 Feature Extraction

Text data is unstructured and unsuitable for machine learning algorithms[16]. This paper uses feature extraction, specifically TF-IDF, to structure and optimize data for machine learning. TF-IDF is a widely used method for determining the relevance of a phrase in a document or series for information retrieval and text mining[17]. It consists of three components:

Term Frequency (TF) is the frequency with which a phrase appears in a document, calculated by dividing the total number of representations by the number of times a word occurs. It is based on the idea that the more frequently a phrase appears, the more essential it is. The formula for calculating TF is as follows: equation 3.1.

$$\text{TF-IDF}(T,D) = \frac{\textit{number of times phrase T appears in document D}}{\textit{Entire number of term in document D}} \quad (1)$$

Where T is a term in Document D.

Inverse Document Frequency evaluates the relevance of words in a corpus of documents by calculating the logarithmic of the document amount divided by the number of records containing the phrase. This component criminalizes less valuable or helpful terms in identifying texts. The formula is described in equation 2.

$$\text{TF-IDF}(T,D) = \frac{\textit{Total number of documnts in corpus D}}{\textit{number of documents containg phrase T}} \quad (2)$$

Where T is a term in document corpus D.

Lastly, combining the Term Frequency and Inverse Term Frequency will form is calculated by multiplying both TF and IDF, which describe in equation 3:

$$\text{TF-IDF. TF-IDF} \quad (3)$$

3.4 Feature selection

Feature selection (FS) is crucial for speeding up text classification models by removing irrelevant features and reducing dataset dimensionality. The ATCM model uses chi-square feature selection for simplicity and speed, calculated using equation 4.

$$\mathbf{chi-square}\ (\mathbf{t}, \mathbf{ci}) = \frac{(\ (\mathbf{N} \times\ \mathbf{AB} + \mathbf{CD}))}{(\mathbf{A} + \mathbf{C}\ \mathbf{B} + \mathbf{D}\ \mathbf{A} + \mathbf{B}\ \mathbf{C} + \mathbf{D})} \tag{4}$$

Suppose Ci is a category in setc1, c2,..., c|C|, and t is a token linked with one or more documents in the training set. N represents the total number of training documents; A represents the number of documents in class ci which contain the token t; B represents the number of documents in other classes which contain the token t; C represents the number of documents in class ci that do not contain the token t; and D represents the number of documents in other classes that do not contain the token t [18].

Because the chi-square FS is originally designed for the traditional classification problem, the multi-label text classification issue needs to adapt feature selection to deal with this problem. ATCM performed feature selection separately for each label and selected the features based on their relevance to that specific label.

Finally, the average score is calculated for each feature across all labels. then the top-ranked features are selected (1000,2000,3000,4000, and 5000) are selected based on their average score. These selected features are used for evaluating the model.

3.5 Multi-label text classification methods

Because documents can belong to several labels at the same time, the problem transformation technique is utilized to turn multi-label Because documents can belong to several labels simultaneously, the problem transformation technique is utilized to turn multi-label classification problems into single ones. For this reason, adopting a problem transformation (PT) approach.

The ATCM focuses on employing binary relevance (BR) and label power set (LP) as PT approaches before using traditional machine learning algorithms. A multi-label classification problem is frequently divided into L labels in a dataset using the BR approach, which then predicates data for each label using binary classification training per label. A multi-label problem is transformed into a multi-class problem via LP. For each conceivable label combination, it takes into account a different class before training a multi-class classifier on this new class dataset[19][20].

3.6 classification algorithms

Selecting the best algorithm is the most essential step of the process for text classification. In this section, ATCM three most common classical machine learning algorithms are employed to classify the RTANews dataset, including Multinomial Naïve Bayes (MNB)[1], Random Forest (RF)[16][21].and Logistic Regression (LR)[22].

3.7 Evolution Metrics

Testing multi-label classifications is more complex than testing single-label classifiers, as multi-label predictions are sets of labels. Traditional evaluation metrics like precision, recall, and F-measure need to accurately reflect the concept, which is partially correct. Two escalation types were used for multi-label classification: example-based and label-based metrics[15]. Example-based metrics evaluate each test instance and average the results of overall test instances, while label-based metrics evaluate each label individually and average the results across all labels[23]. The proposed model uses both approaches to gain a comprehensive understanding of the model's performance, with hamming loss being the first approach and micro-averaged metrics like precision, recall, and F-measure.

a. Example-based metric

Hamming loss is a metric used to assess the effectiveness of multi-label models, determining the proportion of incorrectly predicted labels per example or instance.

$$Hamming\ Loss = \frac{1}{s}\sum_{}^{s}\frac{1}{n}|Zi\ \Delta\ Yi\ | \tag{5}$$

Where s represents the total number of instances in i test set where (1≤ I ≤ s), Yi represents the actual label set for the i-th example, whereas Zi indicates the predicted label set for the i-th example. Δ is a symmetric difference between the actual and predicted label sets represented. The hamming loss ranges between [0,1], Where a value of 1 means that there are no labels predicted correctly, on the other hand, a value of 0 represents all labels predicted correctly[24].

b. Micro-averages metric

Micro-averaged metrics, including Micro Recall, Micro Precision, and Micro F-measure, are metrics for performance that take the entire number of True Positives (TP), False Positives (FP), and False Negatives (FN) in a multi-label set into account. These metrics provide an overall evaluation of the categorization system's performance across all categories, assigning equal weight to each document[25]. Micro-average Precision(P) is calculated as:

$$Micro - averge - \text{Precision}(R) = \frac{\sum Tpi}{\sum Tpi + Fpi} \quad (6)$$

The equation (7) introduced the Micro-average Recall :

$$Micro - averge - Recall \ \ (R) = \frac{\sum Tpi}{\sum Tpi + FNi} \quad (7)$$

Where TPi represents a True Positives for each category i, and FPi ,FNi represents a False Positives and False Negative respectively for each category i. The harmonic mean of equation (3.6) an equation (3.7) are calculated as equation(3.8) below which represent the Micro-average F-measure:

$$Micro - average\ Fmesure = \frac{2(P * R)}{(P + R)}$$

4. Experimental Results and Discussion

4.1 Experimental Setup

The proposed model is created using a laptop that has the specification: Intel i7-11800 H (11 Gen)2.4 GHZ processor, 15.7 GB Available RAM , 512 GB Available Disk and window 11 home operating system . In order to construct the model, Python is used as a programming language. The Jupyter Notebook is used to build up the setting. To train and evaluate model the model. The proposed model utilized the sci-kit-learn package extensively.

4.2 Result and Discussion

This section focuses on evaluating the results of the ATCM model as well as comparing the findings to earlier models' results in the literature, as detailed in Section 2 .to compare, the same Data set utilized in prior research examples is employed.

To evaluate the transformation-based methods, single-label base learners are utilized. Going forward, we refer to the transformation methods and their base learners using an abbreviation format. This format combines the transformation-based method's abbreviation with the base learner's abbreviation, separated by a hyphen symbol. As an example, the abbreviation "BR-LR," refers to "Binary Relevance" as the transformation-based method and "Logistic Regression" as the base learner. The chi-square is used for feature selection, and several subsets of training features, specifically 1000, 2000, 3000, 4000, and 5000 features, were used to evaluate RTAnews. There were six trials conducted, one for each approach, on each set of features. Thus, a total of 30 experiments were run across all feature sets and techniques. All trials were assessed using a thorough set of metrics developed expressly for multi-label learning. One example-based metric is Hamming loss. In addition, three label-based measures were used: micro-averaged Precision, micro-averaged recall, and micro-averaged F-measure. These measures are detailed in Section 3.7. While each evaluation of performance measure assesses performance in a different way than the others, we show and discuss the results of each independently.

a. Hamming Loss

Lower numbers indicate higher performance according to the Hamming Loss metric, which measures label prediction accuracy. LP-LR is the top performer in the table 1, consistently achieving the lowest Hamming Loss across various feature options. Particularly with values of 0.855 (5000 features), 0.0858 (4000 features), and 0.0866 (2000 features), LP-RF performs exceptionally well. With a Hamming Loss of 0.0901 for the first 2000 features and 0.0914 for the last 4,000, LP-RF also exhibits accuracy. This highlights LP-LR's leadership's inaccurate label prediction and LP-RF's competitive performance.

TABLE 1 ATCM RESULTS

	Hamming Loss	Micro-Precision	Micro-Recall	Micro F measure	Features		Hamming Loss	Micro-Precision	Micro-Recall	Micro F measure	Features
BR-LR	0.1075	0.7646	0.8633	0.8109	1000 selected features	BR-LR	0.1027	0.7764	0.8642	0.8180	2000 selected features
BR-RF	0.0894	0.8519	0.8050	0.8278		BR-RF	0.0898	0.8555	0.7989	0.8262	
BR-MNB	0.1088	0.8120	0.7710	0.7910		BR-MNB	0.1067	0.8103	0.7842	0.7970	
LP-LR	0.0911	0.8276	0.8322	0.8299		LP-LR	0.0875	0.8353	0.8375	0.8364	
LP-RF	0.0902	0.8306	0.8320	0.8313		LP-RF	0.0901	0.8306	0.8322	0.8314	
LP-MNB	0.1208	0.7692	0.7821	0.7756		LP-MNB	0.1172	0.7780	0.7851	0.7815	
BR-LR	0.1001	0.7832	0.8646	0.8219	3000 selected features	BR-LR	0.0988	0.7871	0.8636	0.8236	4000 selected features
BR-RF	0.0910	0.8544	0.7949	0.8236		BR-RF	0.0916	0.8554	0.7909	0.8219	
BR-MNB	0.1062	0.8098	0.7875	0.7985		BR-MNB	0.1063	0.8096	0.7870	0.7982	
LP-LR	0.0866	0.8364	0.8398	0.8381		LP-LR	0.0858	0.8389	0.8403	0.8396	
LP-RF	0.0918	0.8270	0.8297	0.8284		LP-RF	0.0914	0.8281	0.8302	0.8291	
LP-MNB	0.1184	0.7757	0.7833	0.7795		LP-MNB	0.1210	0.7704	0.7788	0.7746	
5000 selected features											
BR-LR	0.0984	0.7891	0.8621	0.8240		LP-LR	0.0855	0.8385	0.8418	0.8402	
BR-RF	0.0925	0.8544	0.7878	0.8198		LP-RF	0.0920	0.8264	0.8297	0.8280	
BR-MNB	0.1062	0.8112	0.7847	0.7978		LP-MNB	0.1238	0.7649	0.7742	0.7695	

b. Averaged micro precision

Micro-Precision, a crucial metric for assessing positive label prediction precision, is examined in the table 1. Analyzing the results reveals distinct patterns among different classifiers and feature sets.

Among "BR-LR," the Micro-Precision starts at 0.7646 (1000 features) and improves to 0.7891 (5000 features). Comparatively, "BR-RF" shows higher precision with 0.8519 (1000 features) and 0.8555 (2000 features). "BR-MNB" begins at 0.8120 (1000 features) and slightly drops to 0.8103 (2000 features). In contrast, "LP-LR" maintains consistently high precision: 0.8276 (1000 features), 0.8353 (2000 features), 0.8364 (3000 features), and 0.8389 (4000 features). Similarly, "LP-RF" demonstrates stable performance with Micro-Precision of 0.8306 (1000 features) and 0.8306 (2000 features). "LP-MNB" starts with 0.7692 (1000 features) and increases to 0.7780 (2000 features). Notably, "BR-RF" and "LP-LR" consistently achieve high positive label prediction precision across various feature selections, while "BR-MNB" and "LP-MNB" exhibit slightly lower rates.

In summary, Micro-Precision underscores the accuracy of positive label predictions. "BR-RF" and "LP-LR" shine as consistent performers, while "BR-MNB" and "LP-MNB" show comparatively lower precision..

c. Averaged micro Recall

Micro-Recall, a crucial metric assessing the sensitivity of classifiers in correctly identifying positive instances, is examined within the provided table 1. An exploration of the results unveils distinct trends among various classifiers and feature sets. Starting with the "BR-LR" classifier, Micro-Recall values show how accurately it identifies positive instances. In the context of 1000 features, Micro-Recall is 0.8633, indicating that it correctly identifies around 86.33% of actual positive instances. As the number of features increases to 2000, Micro-Recall remains relatively stable but slightly decreases to 0.8642 (86.42%). Comparing with the "BR-RF" classifier, there is a notable difference in Micro-Recall. With 0.8050 (1000 features) and 0.7989 (2000 features), "BR-RF" demonstrates a comparatively lower sensitivity in correctly identifying positive instances. Transitioning to the "BR-MNB" classifier, the Micro-Recall values indicate its effectiveness in identifying positive instances. Starting at 0.7710 (1000 features), it slightly improves to 0.7842 (2000 features), showcasing its capability to correctly identify more actual positive instances. Shifting focus to the "LP-LR" classifier, Micro-Recall emphasizes its high sensitivity in accurately identifying positive instances. With values of 0.8322 (1000 features), 0.8375 (2000 features), 0.8398 (3000 features), and 0.8403 (4000 features), it maintains a consistently robust performance across various feature selections. In the case of "LP-RF," Micro-Recall starts at 0.8320 (1000 features) and remains stable with 0.8322 (2000 features), showcasing its reliability in identifying positive instances. Finally, the "LP-MNB" classifier, with Micro-Recall values of 0.7821 (1000 features) and 0.7851 (2000 features), demonstrates its proficiency in correctly identifying positive instances.

In summary, the Micro-Recall metric reveals the classifiers' sensitivity in identifying positive instances. "BR-LR" and "LP-LR" classifiers consistently excel in this aspect, while "LP-RF," "BR-MNB," and "LP-MNB" also demonstrate commendable sensitivity. Comparatively, "BR-RF" exhibits a comparatively lower rate of correct identification of positive instances.

d. Averaged micro F-measure

Micro F-measure, a significant metric combining precision and recall, offers a comprehensive evaluation of classifier performance. Delving into the provided table 1 reveals distinct trends among different classifiers and feature sets, shedding light on their overall effectiveness .Examining the "BR-LR" classifier, Micro F-measure values highlight its balanced performance in trade-offs between precision and recall. With a value of 0.8109 (1000 features), it demonstrates an equilibrium between the two metrics. As features increase to 2000, the Micro F-measure slightly improves to 0.8180, sustaining its balanced nature. Comparing with the "BR-RF" classifier, a higher overall performance is apparent. It achieves a Micro F-measure of 0.8278 (1000 features) and maintains this heightened performance with 0.8262 as features increase to 2000.The "BR-MNB" classifier showcases balanced performance as well. Starting with a Micro F-measure of 0.7910 (1000 features), it improves marginally to 0.7970 (2000 features), maintaining its equilibrium between precision and recall. Turning to the "LP-LR" classifier, it demonstrates commendable overall performance with Micro F-measure values of 0.8299 (1000 features), 0.8364 (2000 features), 0.8381 (3000 features), and 0.8396 (4000 features). This consistency underscores its balanced effectiveness in combining precision and recall. "LP-RF" maintains stable Micro F-measure values, starting at 0.8313 (1000 features) and maintaining this level with 0.8314 (2000 features), showcasing its consistent balanced performance. "LP-MNB," while slightly lower in overall performance, still exhibits competitive Micro F-measure values. It starts at 0.7756 (1000 features) and decreases slightly to 0.7815 (2000 features).

In summary, Micro F-measure provides a comprehensive perspective on classifier performance, encapsulating both precision and recall. "BR-RF" and "LP-LR" consistently excel with strong overall performance, while "BR-LR," "BR-MNB," and "LP-RF" also maintain balanced and competitive values. "LP-MNB" demonstrates slightly lower performance, but overall, the classifiers showcase effective performance in combining precision and recall..

5. Conclusion and future work

This paper introduced a new model for Multi-label as solution for the multi-label Arabic language of textual documents dataset called "RTANEWS" . The proposed model "ATCM" improved the performance of the multi-label system . This paper focused on two approaches : Binary Relevance and Label power set technique to handle multi-label problem. From the experimental results, many conclusions are derived including : A new model for Arabic text

based on the Arabic documents dataset is presented in this work and given better results for increasing accuracy by proposing a text classification method. For future works

1. Utilizing Deep Learning for Multi-label text Classification in Arabic such as ARBERT improve performance in classifying multiple categories.

2. Expanding Techniques for Multi-label Texts: use other techniques such as classifier chain and adaption techniques handling multi-label texts in Arabic.

3. Enhance text classification and understanding by expanding the machine learning algorithms used.

4. Utilizing languages other than Arabic for model generalization to generalize the proposed model, including :Persian, Amharic, English, etc.

References

[1] A. H. Mohammad, "Arabic Text Classification: A Review," *Mod. Appl. Sci.*, vol. 13, no. 5, p. 88, 2019, doi: 10.5539/mas.v13n5p88.

[2] M. Abbas, K. A. Memon, A. A. Jamali, S. Memon, and A. Ahmed, "Multinomial Naive Bayes Classification Model for Sentiment Analysis," *IJCSNS Int. J. Comput. Sci. Netw. Secur.*, vol. 19, no. 3, p. 62, 2019, doi: 10.13140/RG.2.2.30021.40169.

[3] F. Peters, T. T. Tun, Y. Yu, and B. Nuseibeh, "Text Filtering and Ranking for Security Bug Report Prediction," *IEEE Trans. Softw. Eng.*, vol. 45, no. 6, pp. 615–631, 2019, doi: 10.1109/TSE.2017.2787653.

[4] A. Elsaid, A. Mohammed, L. F. Ibrahim, and M. M. Sakre, "A Comprehensive Review of Arabic Text Summarization," *IEEE Access*, vol. 10, pp. 38012–38030, 2022, doi: 10.1109/ACCESS.2022.3163292.

[5] N. Aljedani, R. Alotaibi, and M. Taileb, "Multi-Label Arabic Text Classification: An Overview," 2020.

[6] S. Kumar, N. Kumar, A. Dev, and S. Naorem, "Movie genre classification using binary relevance, label powerset, and machine learning classifiers," *Multimed. Tools Appl.*, 2022, doi: 10.1007/s11042-022-13211-5.

[7] M. K. B. Melhem, L. Abualigah, R. A. Zitar, A. G. Hussien, and D. Oliva, *Comparative Study on Arabic Text Classification: Challenges and Opportunities*, vol. 1071. Springer International Publishing, 2023. doi: 10.1007/978-3-031-17576-3_10.

[8] M. A. R. Abdeen, S. AlBouq, A. Elmahalawy, and S. Shehata, "A closer look at arabic text classification," *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 11, pp. 677–688, 2019, doi: 10.14569/IJACSA.2019.0101189.

[9] N. A. Ahmed, M. A. Shehab, M. Al-Ayyoub, and I. Hmeidi, "Scalable multi-label Arabic text classification," *2015 6th Int. Conf. Inf. Commun. Syst. ICICS 2015*, pp. 212–217, 2015, doi: 10.1109/IACS.2015.7103229.

[10] A. Y. Taha and S. Tiun, "Binary relevance (BR) method classifier of multi-label classification for arabic text," *J. Theor. Appl. Inf. Technol.*, vol. 84, no. 3, pp. 414–422, 2016.

[11] M. A. Shehab, O. Badarneh, M. Al-Ayyoub, and Y. Jararweh, "A supervised approach for multi-label classification of Arabic news articles," *Proc. - CSIT 2016 2016 7th Int. Conf. Comput. Sci. Inf. Technol.*, pp. 1–6, 2016, doi: 10.1109/CSIT.2016.7549465.

[12] I. Hmeidi, M. Al-Ayyoub, N. A. Mahyoub, and M. A. Shehab, "A lexicon based approach for classifying Arabic multi-labeled text," *Int. J. Web Inf. Syst.*, vol. 12, no. 4, pp. 504–532, 2016, doi: 10.1108/IJWIS-01-2016-0002.

[13] B. Al-Salemi, M. Ayob, G. Kendall, and S. A. M. Noah, "Multi-label Arabic text categorization: A benchmark and baseline comparison of multi-label learning algorithms," *Inf. Process. Manag.*, vol. 56, no. 1, pp. 212–227, 2019, doi: 10.1016/j.ipm.2018.09.008.

[14] R. M. Al Mgheed, "Scalable Arabic text Classification Using Machine Learning Model," *2021 12th Int. Conf. Inf. Commun. Syst. ICICS 2021*, pp. 483–485, 2021, doi: 10.1109/ICICS52457.2021.9464566.

[15] N. Aljedani, R. Alotaibi, and M. Taileb, "HMATC: Hierarchical multi-label Arabic text classification model using machine learning," *Egypt. Informatics J.*, vol. 22, no. 3, pp. 225–237, 2021, doi: 10.1016/j.eij.2020.08.004.

[16] K. Kowsari, K. J. Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown, "Text classification algorithms: A survey," *Inf.*, vol. 10, no. 4, pp. 1–68, 2019, doi: 10.3390/info10040150.

[17] C. Puttipornchai, S. Chanyachatchawan, and N. Tuaycharoen, "Multi-Label Classification for Articles in Thai Journal Database from Article's Abstract," *2022 19th Int. Jt. Conf. Comput. Sci. Softw. Eng. JCSSE 2022*, pp. 1–6, 2022, doi: 10.1109/JCSSE54890.2022.9836270.

[18] D. S. Guru, M. Ali, M. Suhil, and M. Hazman, "A study of applying different term weighting schemes on Arabic text classification," *Lect. Notes Networks Syst.*, vol. 43, pp. 293–305, 2019, doi: 10.1007/978-981-13-2514-4_25.

[19] D. Ganda, R. B.-R. T. in P. Languages, and undefined 2018, "A survey on multi label classification," *Researchgate.Net*, vol. 5, no. 1, pp. 19–23, 2018, [Online]. Available: https://www.researchgate.net/profile/Rachana-Buch/publication/327110772_A_Survey_on_Multi_Label_Classification/links/5bf56905299bf1124fe4aef2/A-Survey-on-Multi-Label-Classification.pdf

[20] R. A. Zayed, M. F. A. Hady, and H. Hefny, "Islamic fatwa request routing via hierarchical multi-label Arabic text categorization," *Proc. - 1st Int. Conf. Arab. Comput. Linguist. Adv. Arab. Comput. Linguist. ACLing 2015*, pp. 145–151, 2016, doi: 10.1109/ACLing.2015.28.

[21] H. N. Alshaer, M. A. Otair, L. Abualigah, M. Alshinwan, and A. M. Khasawneh, "Feature selection method using improved CHI Square on Arabic text classifiers: analysis and application," *Multimed. Tools Appl.*, vol. 80, no. 7, pp. 10373–10390, 2021, doi: 10.1007/s11042-020-10074-6.

[22] S. T. Indra, L. Wikarsa, and R. Turang, "Using logistic regression method to classify tweets into the selected topics," *2016 Int. Conf. Adv. Comput. Sci. Inf. Syst. ICACSIS 2016*, no. August 2017, pp. 385–390, 2017, doi: 10.1109/ICACSIS.2016.7872727.

[23] F. Elghannam, "Multi-Label Annotation and Classification of Arabic Texts Based on Extracted Seed Keyphrases and Bi-Gram Alphabet Feed Forward Neural Networks Model," *ACM Trans. Asian Low-Resource Lang. Inf. Process.*, vol. 22, no. 1, 2022, doi: 10.1145/3539607.

[24] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Mining Multi-label Data," pp. 1–20.

[25] S. S. Sonawane, P. N. Mahalle, and A. S. Ghotkar, "Information Retrieval," *Stud. Big Data*, vol. 104, pp. 81–94, 2022, doi: 10.1007/978-981-16-9995-5_4.