Hindawi
Mathematical Problems in Engineering
Volume 2022, Article ID 3720358, 16 pages
https://doi.org/10.1155/2022/3720358

Research Article

Arabic Document Classification: Performance Investigation of Preprocessing and Representation Techniques

Abdullah Y. Muaad,[1,2] **Hanumanthappa Jayappa Davanagere**,[1] **D.S. Guru**,[1] **J.V. Bibal Benifa**,[3] **Channabasava Chola**,[3] **Hussain AlSalman**,[4] **Abdu H. Gumaei**,[5] **and Mugahed A. Al-antari**[6]

[1]*Department of Studies in Computer Science, University of Mysore, Manasagangothri, Mysore-570006, India*
[2]*Sana'a Community College, Sana'a 5695, Yemen*
[3]*Department of Computer Science and Engineering, Indian Institute of Information Technology, Kottayam, India*
[4]*Department of Computer Science, College of Computer and Information Sciences, King Saud University, Riyadh, 11543, Saudi Arabia*
[5]*Computer Science Department, Faculty of Applied Sciences, Taiz University, Taiz 6803, Yemen*
[6]*Department of Artificial Intelligence, Daeyang AI Center, Sejong University, Seoul 05006, Republic of Korea*

Correspondence should be addressed to Abdullah Y. Muaad; abdullahmuaad9@gmail.com, Hanumanthappa Jayappa Davanagere; hanumsbe@gmail.com, Abdu H. Gumaei; abdugumaei@gmail.com, and Mugahed A. Al-antari; en.mualshz@sejong.ac.kr

Received 5 October 2021; Revised 11 March 2022; Accepted 6 April 2022; Published 30 April 2022

Academic Editor: Dost Muhammad Khan

With the increasing number of online social posts, review comments, and digital documentations, the Arabic text classification (ATC) task has been hugely required for many spontaneous natural language processing (NLP) applications, especially within the coronavirus pandemics. The variations in the meaning of the same Arabic words could directly affect the performance of any AI-based framework. This work aims to identify the effectiveness of machine learning (ML) algorithms through preprocessing and representation techniques. This effectiveness is measured via different AI-based classification techniques. Basically, the ATC process is influenced by several factors such as stemming in preprocessing, method of feature extraction and selection, nature of datasets, and classification algorithm. To improve the overall classification performance, preprocessing techniques are mainly used to convert each Arabic word into its root and decrease the representation dimension among the datasets. Feature extraction and selection always play crucial roles to represent the Arabic text in a meaningful way and improve the classification accuracy rate. The selected classifiers in this study are performed based on various feature selection algorithms. The overall classification evaluation results are compared using different classifiers such as multinomial Naive Bayes (MNB), Bernoulli Naive Bayes (BNB), Stochastic Gradient Descent (SGD), Support Vector Classifier (SVC), Logistic Regression (LR), and Linear SVC. All of these AI classifiers are evaluated using five balanced and unbalanced benchmark datasets: BBC Arabic corpus, CNN Arabic corpus, Open-Source Arabic corpus (OSAc), ArCovidVac, and AlKhaleej. The evaluation results show that the classification performance strongly depends on the preprocessing technique, representation methods and classification technique, and the nature of datasets used. For the considered benchmark datasets, the linear SVC has outperformed other classifiers overall when prominent features are selected.

1. Introduction

The concept of text mining (knowledge discovery) has gained much attention in recent years to extract meaningful information from the text [1]. It is a more recent discipline that emerged from the fields of linear algebra, statistics, ML, natural language processing (NLP), and linear algebra [2]. In addition, text mining has been categorized as a big data problem because of the volume, variety, veracity, and velocity of data [3]. Text mining techniques can mine/discover

information from huge datasets, and on the other hand, they allow flexibility and adaptability. This analysis helps in many domains such as classification of text data in many real scenarios such as classification of articles, e-mail spam classification, text translation, and sentiment analysis. Almost 447 million people speak Arabic as a native language, and it is declared as the official language for 22 countries across the globe. Moreover, it is the mother script for other languages such as Persian and Urdu. The Arabic language comprises 29 letters/characters, and typically, the writing for Arabic is opposite to English from the right side to the left side. One of the main characteristics of the Arabic text is that the letters are position-dependent with different forms and shapes [4]. In comparison to other languages, Arabic is a root-based language, and the majority of Arabic words consist of vertical stacking of letters. Besides that, Arabic has three different forms known as classical, standard, and dialectal. Therefore, working on Arabic text mining is a difficult task as compared to other languages. In addition, the Arabic language is more challenging to learn because of the dialects, phonology, orthography, and morphology characteristics [5]. Further, only limited research works have been done in ATC in comparison with the English language [6].

Data preprocessing is the first practice of preparing and cleaning the text for further processing. It is the first phase in the text classification pipeline before representation because the text is often unstructured. The unstructured text does not have a predefined data model, and it is not suitable for further processing [7]. Therefore, specific algorithms for preprocessing are required to reduce the text size, eliminate noise from the text, and extract useful patterns. Mainly, preprocessing also includes text cleaning, space, and stop words removal, stemming, and handling of negation words [8]. Preprocessing text plays the main role in enhancing the accuracy of Arabic documents in many applications [9]. Finally, most of the accuracy results are affected by preprocessing difficulties [10].

The Arabic text representation and feature extraction are a seriously important step in the text mining and classification process to extract the most prominent features and select the optimal subset. It is a process of representing text or documents as vectors and transforming from unstructured to structured equivalent such that the ML algorithms can understand it precisely [11]. Several feature extraction techniques were applied such as BoW, TF-IDF [12], Term-Class Relevance TCR, Term Class Weight-Inverse Class Frequency (TCW-ICF) [3], phrases representation [13], and a symbolic used for feature representation [14]. Most of the researches use TF-IDF or BoW, which have some major feature dimensionality reduction (FDR) drawbacks like missing the order of the words, and they also ignore semantics of the words so that different sentences can have the same representation, as long as the same words are used. TF-IDF and BoW techniques are usually used as representation models but still, they have limitations such as sparse matrices. Therefore, the way of factorization and FDR is inevitable to finalize the relatively small number of features for every document [3]. FDR technique can be done in the preprocessing step such as stop word removal. Feature Selection (FS) is done by any of the three methods known as embedded, wrapper, and filtering techniques. Feature transformation is one of the FDR techniques that belong to the category of multivariate analysis. Principal Component Analysis (PCA), Nonnegative Matrix factorization (NMF), Random Projection (RP), and Linear Discriminant Analysis (LDA) are used for unsupervised feature extraction process [15]. Based on the observations, these methods can be applied effectively for decreasing the size of the representation matrix in Arabic text documents. Once the representation of text is created through optimal feature extraction and selection methods, then the classifier algorithm has to be trained. Subsequently, the trained model should classify the documents into target classes with improved accuracy, based on the knowledge gained at the training phase [3]. Text categorization is a significant task that detects whether a piece of writing text belongs to any of the predefined sets of classes [16]. In addition, text categorization remains one of the most difficult computational tasks in the ML field [17]. There are several applications of text mining, such as spam and junk e-mail filtering, Web page classification, and sentiment analysis. Finally, compared to English, a few researches have been done for ATC by NB [18], SVM [19], ANN [20], and DT [21].

The goal of this study is to investigate the impact of preprocessing and representation of various techniques on the Arabic text classification. In addition to these feature extraction and selection methods, six AI-based classifiers are adopted and employed for the Arabic text classification task. Consequently, this research enables the developers in the domain to select a robust AI-based technique for the robust ATC applications. The main contributions of this study are summarized as follows:

(i) A comprehensive study is performed to investigate the effectiveness of Arabic text classification workflow procedure: preprocessing, representation, and classification.

(ii) Several AI-based techniques are adopted and tested along with different preprocessing algorithms to identify their performance with respect to ATC.

(iii) To show the effectiveness of preprocessing and representation techniques, all AI-based classifiers are evaluated via five Arabic text benchmark datasets, namely, CNN Arabic (CNN), BBC Arabic (BBC), Open-Source Arabic Corpus (OSAc), ArCovidVac, and AlKhaleej.

(iv) We show the effectiveness of balanced and unbalanced Arabic datasets in multiclass classification scenarios.

The rest of the article is structured as follows. In Section 2, the related previous works are summarized regarding English and Arabic language text classification. In Section 3, the proposed methodology and the mathematical model are discussed. Then, the detailed experimental analysis and the outcomes are described in Section 4. The key findings and the inferences are detailed in Section 5 with appropriate concluding remarks.

2. Related Work

In English and other languages, many different approaches are available to solve the problem of text classification. However, only a sparse amount of research works has reported the key issues involved in the Arabic text classification (ATC). In recent days, the revolution created around the globe by social media and search engines has stressed the need for ATC in terms of Arabic text mining, sentimental analysis, and document classification. Here, a survey of related works is addressed to characterize the challenges involved in the pipeline of text classification as stated in the introduction part of the article. The ATC process consists of four different phases, namely, prepossessing, representation, FDR, and classification.

2.1. Text Preprocessing. There is a collection of preprocessing methods such as stop words, which are used to eliminate unwanted words with high weights. For example, a, an, and the do not have significant meaning; therefore, deleting these words will not affect the model but will decrease the dimension like the authors in [1] have done for English. In contrast, the same idea has been used for the removal of Arabic text stop words such as [11] [2] [22] جل ، مع، لىال. Stemming is the method of converting a word into its corresponding root or stem. Thus, it is seen as an important preprocessing step before handling any NLP tasks. Stemming is a very important process for Arabic text mining. Primarily, there are three types of stemmers known as root-based (Khoja), light-based (Light 10), and statistical-based (N-Grams) stemmers, and occasionally they will be further subdivided into many other types. It is important to apply the steaming process to the text data after tokenization. In Arabic text, Yousif et al. (2015) evaluated the effect of stemming algorithm along with NB classifier and concluded that the root extractor exhibits the best performance [23]. Previously, Rehab Duwairi et al. (2013) investigated the effects of stemming strategies on Arabic document classification [24]. They used a stemming algorithm called light stemming, and word clusters were investigated, and they conclude that the light stemming method improves the result in terms of accuracy. Mamoun and Ahmed (2016) proposed a new light stemmer for Arabic called P-stemmer. They modified the version of Larkey's light stemmers. The validation of the stemmer has shown significantly increasing accuracy during the classification process when NB, SVM, and RF classifiers are applied. They conclude that the SVM classifier performs better than other classifiers [4]. Shargabi et al. (2011) used many classification algorithms such as NB, KNN, and J48 to build the classification models, and they conclude that the Light stemmer was better compared to Khoja stemmer [25]. Salam et al. (2016) presented the ATC system by studying the effect of normalization and stemming techniques such as ISRI and Tashaphyne stemmer. Finally, they concluded that the normalization was best for the outcome, but the stemmers could not produce the required accuracy [26]. Oraby et al. (2013) presented Arabic sentiment analysis, and they addressed the effect of the stemming algorithm. They showed the results of accuracy equal to 93.2%, 92.6%, and 92.2% for Tashaphyne, ISRI, and Khoja stemmers, respectively [27]. Qusay Walid et al. (2005) processed a document clustering process by studying the influence of ISRI stemmer [28]. They have concluded that the method of ISRI outperforms nonstemming methods. Al haj et al. (2018) introduced responsibilities of the stemming technique for Arabic text mining, which have been explored, and it has been found that stemming minimizes the computational difficulties for the classification process [11].

2.2. Text Representation. The text cannot be processed in the native form, because it is unstructured. The term unstructured is a very subjective property of text data, so there are many representation schemes available to address this problem with its own set of characteristics. The representation schemes are broadly classified into two types: the first is vector space-based representation model (VSM), and the second is a graph-based representation (GBR) [3].

The VSM is a representation method that has been used because it provides a matrix representation for the text data. According to Sebastiani et al. (2002), ML techniques can be utilized for the process of text classification. There are different types of VSMs present depending on the usage of BoW, in which the words are present in a linearized way [18]. It can be used as features/terms to represent the text data in the form of a vector [12, 14]. Hence, a document-term matrix is created using the vectors corresponding to each document. This representation has limitations like lack of correlation among sentences, and it misses the contextual meaning of keywords in the text document [29].

In contrast, text-based phrases carry high semantic qualities; however, they lead to the loss of statistical quality as Hammouda and Kamel (2004) mentioned in [13]. Kyriakopoulou et al. (2006) applied clustering to the documents followed by a simple BoW model [30]. Term weighting schemes represent the text in the form of a document-term matrix such as TF-IDF. The different variants have unsupervised weighting schemes, which are similar to the IR fields. Elmasry et al. (2018) applied the TCW-ICF for ATC and obtained comparatively better results than TF-IDF [31]. In ATC, the ontology of semantics model has been presented across the words of a text document [32]. As compared to the VSM, the documents are represented in the form of a graph in GBR to capture such correlating words with an appropriate weight. Here, the connected ontology representation preserves the semantic meaning among the words in a text. Guru et al. (2010) proposed a novel symbolic text clustering method along with a symbolic feature selection method [33]. Syntactic Word Representation such as N-gram and other techniques mentioned by Grigori Sidorov et al. (2019) in [34] consist of sparse levels. In contrast, due to the deep learning revolution, many other techniques are also proposed like word embedding with variants by Earlier, Rong (2014). They proposed the Word2Vec method using Skipping N-gram and Continuous BoW (CBoW) technique, but it has failed to capture the contextual meaning of the word from the document, and it could not capture the word

that is not present in the training phase, namely, out-of-vocabulary [35]. Bojanowski (2017) proposed the fast text approach that cannot capture the word that has a different meaning (polysemy). It needs huge memory for storing and pretraining purposes in comparison with GloVec [36] and Word2Vec [37]. Matthew E. Peters et al. (2018) propounded a new idea for representing text with handling contextualized meaning, but the required memory for storage and the performance of downstream tasks are also found computationally more expensive. It could not handle OOV words from the corpus and did not work at a character and word level [38]. Earlier, Mesleh et al. (2007) addressed the enhancement of two metrics precision and recall values by the usage of feature selection technique for Arabic text [19, 39]. In contrast, once deep learning techniques are proposed for Arabic text mining, high accuracy is achieved as compared to ML techniques using public datasets [40]. M. Alhawarat et al. applied a deep learning model, which got an accuracy range of 97.58% to 99.90%; however, the CNN model takes a longer time to complete the training process as compared to the traditional ML algorithms. Hence, it is evident that the ML techniques have the potential to reduce the training time by adding FDR methods. Elnagar et al. (2020) introduced novel unbiased single and multilabeled datasets for Arabic text categorization known as SANAD and NADiA. These datasets are investigated to identify the impact of word2vec embedded models on the performance improvement of the classification tasks [41]. Fatima et al. (2018) presented an Arabic text classification system depending on BoW, utilizing Arabic word sense [42], deep neural network [43], and deep Autoencoder representations [44]. The proposed method has achieved good results using precision and F-measure of 94% and 93%, respectively. Here, they failed to handle the issue of Arabic language ambiguity that enhances the performance of ATC by sense embedding techniques. A. Y. Muaad et al. in [45] proposed a new model for Arabic text called computer-aided recognition (ArCAR), which represented text in character-level. The same idea was presented in [46].

2.3. Feature Dimensional Reduction (FDR). In the text mining process, thousands of terms are created through the process known as BoW. The major issues in text mining are known as noise, redundancy, and huge numbers of features, and they can be handled effectively by feature extraction techniques. FDR is done by feature selection or feature transformation. It is good to mention that less scalability and high time consumption are the key issues in dimensionality reduction. Hence, the majority of methods use feature selection rather than feature transformation [3]. There are plenty of feature selection techniques available for handling text data such as Chi-square, information gain, mutual information, and document frequency [31]. Nehar et al. (2016) introduced ATC by discussing a method for word root extraction without relying on any dictionary [47]. Here, they removed any non-Arabic letters and stop words, etc. Subsequently, LibSVM is used to build the classification model, and the result mentions that the performance of the classification was increased by root extraction. Bahassine et al. (2018) proposed the improved feature selection (FS), known as ImpCHI, and it is compared with other feature selection algorithms, namely, MI, IG, and CHI. The results indicate that ImpCHI with SVM was the best [48]. Alhaj et al. (2019) studied document classification using FS techniques such as CHI and IG [49]. They also studied different kinds of stemmers known as ISRI, Tashaphyne, and ARLStem along with TF-IDF and Chi-square to select the highest ranked Features. The results show that ARLStem accompanied by SVM offers a good classification performance. Elmasry et al. (2018) proposed a new FS technique by transforming the attributes of the TF-IDF term weighting technique [31]. Mohamed (2020) compared three-dimension reduction algorithms for identifying the pros and cons among each other. They concluded that principal component analysis (PCA) has produced effective results in ATC [50]. Chantar et al. (2019) proposed binary Grey Wolf Optimizer (GWO) as a wrapper-based FS technique. From all analyses of results for the proposed model, the SVM enhances the performance for ATC problems as compared to the other [51].

From this related work, we observed that many researchers have applied classical representation like TFIDF and BoW in ATC to represent the text. Besides these representation methods, we try to reduce the size and enhance the performance using the FDR technique [52]. So, an efficient FDR technique is required to reduce the representation cost.

2.4. Text Classification. Once the representation matrix for a given collection of documents or corpus is created through an optimal set of features, a classifier has to train and classify the documents of different classes. The selection of a classifier is an extremely crucial task in the text classification process [3, 17]. Over the last few decades, text categorization problems have been intensively investigated and handled in a variety of real-world applications. Many researchers are increasingly interested in designing applications that use text categorization algorithms, especially in light of recent advances in NLP and text mining. In the literature of ML, there have been plenty of classifiers proposed including both parametric [17] and nonparametric classifiers [7, 12]. Mesleh et al. (2007) implemented a text classification system for Arabic language articles using SVM for classification and Chi to select features [19]. Mohamed El Kourdi (2013) used KNN and FS techniques with stemming and light stemming in preprocessing to achieve classification purposes. The word is reduced to their root, but using KNN needs to find an optimal k-value, and it is computationally expensive [53]. Al-Harbi et al. (2008) proposed Arabic text documents categorization on seven different Arabic corpora using a statistical technique. It is applied using traditional FS with SVM and C5.0 and concluded that a novel FS and weighting strategy is required for achieving an optimal accuracy [22]. Harrag and El-Qawasmah (2009) utilized the Term Weighting Scheme (TWS), Singular Value Decomposition (SVD), and ANN to classify documents in the Arabic

language. From the literature survey of classification algorithms for Arabic text documents, it is inferred that several research gaps exist as stated in the sections above. In addition, a novel stemmer and FS strategy is mandatory to achieve good ATC performance [20].

3. Material and Method

The proposed framework presented the Arabic text classification model using different preprocessing and representation techniques such as bag of word (BoW) and term frequency-inverse document frequency (TF-IDF). Meanwhile, several classifiers are adopted and performed, such as Multinomial NB, Bernoulli NB, LR, SGD Classifier, SVC, and Linear SVC. The findings of the present study are utilized to understand the influence of different methods in the performance improvement of the ATC system. The high-level workflow of the proposed system includes preprocessing, representation, FS, and classification algorithm as highlighted in Figure 1. The preprocesses of the input text start by removing the stop words. Subsequently, the normalization and stemming process have been done to get the root for that text, so that the dimensions can be reduced.

The dimension-reduced text is passed through TF-IDF and BoW that generates a matrix as input. Then, the machine learning started passing this matrix into the ML algorithm after splitting it into two parts: 80% for training, and 20% for testing.

3.1. Preprocessing. The preprocessing technique is most commonly used for preparing the raw data into a specific input data format that ML models can work with. In document classification techniques, preprocessing method refers to the process of converting documents into a suitable form and making text ready for representation. This method is helpful in reducing the computational complexity [11, 22]. In Arabic text, the preprocessing phase eliminates all characters that do not have significant meaning, stop words, and punctuations [54, 55]. The various steps of preprocessing include (a) tokenization, (b) normalization, (c) stop word elimination, and (d) stemming as follows:

3.1.1. Tokenization. The process of splitting text into tokens and replacing each word by number is known as tokenization (features). In Arabic, sentences are frequently divided by distinct signals such as commas, quotes, semicolons, spaces, and periods. These tokens could be single words (noun, verb, pronoun, etc.) that have been altered without regard for their meanings or relationships [56, 57].

3.1.2. Normalization. Normalization is referred to as the transformation of a letter in the text into a canonical form or to remove the diacritics; for instance, transforms ـه into ة. Reference [46].

3.1.3. Stop Word Elimination. It is the process of removing words in the sentences that do not hold any important meaning, for example, for الحال, which means "so." The researcher in Arabic mentions a list of words in [58, 59].

3.1.4. Stemming. The process of removing most frequently prefixes and suffixes and definite articles from the word is named the steaming process. This process is to make a word in a root/base form. There are many types of stemming techniques available such as root, light [11], and hybrid stemming [59].

3.2. Feature Representation (Extraction). Typically, the text processing cannot be done in its native form, because it is unstructured in nature. Many text feature representation schemes have been discussed already with their own characteristics. Initially, the dataset contains a group of documents with many classes as expressed in (1). Then, the technique for representation in terms of Term Frequency (TF) is presented in (2) displayed in the average number of times that occurs in a specific topic, which is divided by the rate of occurrences.

$$AD = d_1, d_2 \ldots\ldots d_N, \tag{1}$$

$$\mathrm{tf}\,(\mathrm{t}, \mathrm{d}) = \frac{f_{\mathrm{d}}(t)}{\max_{w \in d} f_d(w)}. \tag{2}$$

The $f_d(t)$ is the rate of term 't' in the document 'd' and 'w' is a set of words in the document 'd' and D is the corpus of documents.

The measure of information provided by TF is IDF, to see whether a particular term is frequently or rarely used in all documents (common) that is equal to the logarithm of the quotient divided by the total number of documents as expressed in equation (3) and $tf - idf$ is presented in equation (4).

$$idf\,(t, D) = \ln\left(\frac{|D|}{|\{d \in D; t \in d\}|}\right), \tag{3}$$

$$tf - idf\,(t, d, D) = tf\,(t, d) \cdot idf\,(t, D). \tag{4}$$

Generally, in text classification, thousands of terms are obtained through the process of representation such as BoW creation vector. It is a major issue with course dimension due to the characteristics such as noise, redundancy, and a large number of features. Hence, in the feature extraction process, determination of an efficient technique for representation is a critical phase, and there are many methods and different levels for representation [29, 45].

3.3. Arabic Document Classification. Classification of text documents based on their content is known as document classification. Several works have been reported herein based on text classification-oriented examples such as sentimental analysis, rating classification, and document classification with the help of many classification algorithms, with few well-known algorithms as follows: multinomial NB, Bernoulli NB, stochastic gradient descent (SGD), and logistic

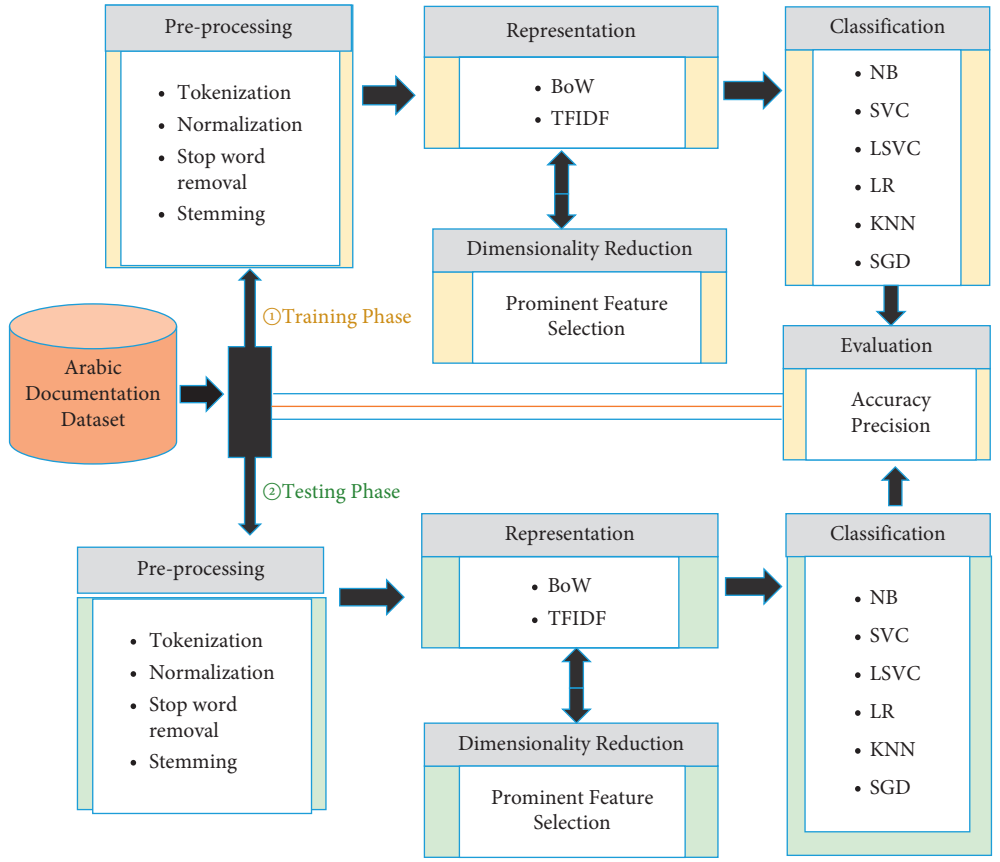

Figure 1: Schematic diagram of the proposed Arabic text classification architecture.

regression [21, 60, 61]. Classifiers have been introduced in recent works. In the proposed work, five open-source datasets, namely, CNN, BBC, OSAC, ArCovidVac, and ALKHALEEJ, are used with 6, 7, 10, and 10 classes, respectively, for document classification in Arabic text. Here, three datasets are unbalanced, and one dataset is balanced. Classification is the labeling of data documents or text to their classes based on their content. In the present work, the classification performance of the classifiers, namely, Multinomial NB [61–63], Bernoulli NB [64], SGD Classifier [65], SVC [66], Linear SVC [67, 68], and logistic regression, is explored, and the appropriate discussions are included in the subsequent sections.

3.3.1. Multinomial Naive Bayes (MNB). Naive Bayes in general is a probabilistic model. It has different forms such as MNB. Based on the explanation of MNB distribution and Bayes' rule in [5[1]3], MNB classifiers use the following formulas in equation (5). The document 'd' belongs to class 'c' that is estimated using equation to get its probability:

$$P(c|d)\alpha P(c)\prod_{1\le k\le n_d} P(t_k|c), \tag{5}$$

where $P(t_k|c)$ is the conditional probability, $P(c)$ is the prior probability, and $\langle t_1, t_2, \ldots t_{n_d}\rangle$ are the tokens in 'd'. n_d is the number of tokens in 'd'.

The aim is to find the best class for the text or document we want to classify. The maximum a posteriori (MAP) c_{map} is the best classification in NB classification

$$c_{\text{map}} = \arg\max_{c\in C} \widehat{P}(c|d) = \arg\max_{c\in C} \widehat{P}(c)\prod_{1\le k\le n_d} \widehat{P}(t_k|c), \tag{6}$$

$$c_{\text{map}} = \arg\max_{c\in C}\left[\log\widehat{P}(c) + \sum_{1\le k\le n_d} \log\widehat{P}(t_k|c)\right], \tag{7}$$

$$\widehat{P}(c) = \frac{N_c}{N};$$
$$\widehat{P}(t|c) = \frac{T_{ct}}{\sum_{t'\epsilon V} T_{ct'}}, \tag{8}$$

where N_c is the number of documents in class c and N is the total number of documents. The conditional probability $\widehat{P}(t|c)$ is the relative frequency of term 't' in documents belonging to class c as in (8). Here, T_{ct} is the number of occurrences of 't' in training documents from class c.

3.3.2. Bernoulli Naive Bayes (BNB). These models are independent binary variables that describe the input $X = \langle 1; 0; 1: : 1\rangle$, which means that the binary term

occurrence is used instead of the frequency of the term in the BoW model as in

$$P(d|c) = P(\langle e_1 \cdots e_M \rangle | c) = \prod_{1 \le i \le M} P(U_i = e_i|c). \quad (9)$$

$U_t = 1$ iff 't' occurs in the document, $d = \langle e_1, e_2, \ldots, e_M \rangle$, where $e_i = e|c$, $\widehat{P}(U_i = e|c)$ and $\widehat{P}(c)\prod_{t_i \in V}\widehat{P}(U_i = e_i|c)$.

$d = \langle e_1, e_2 \ldots, e_M \rangle$ is a binary vector of dimensionality M that indicates, for each term, whether it occurs in d or not, and 'c' is the class.

3.3.3. Logistic Regression. LR is an expanded form of linear regression, which creates discrete probability scores using logistic sigmoid functions. This aids in classification or prediction by generating a regression function. The logistic regression model used binary classification problems, but it can also be expanded to multiclassification situations. In logistic regression, any input value is mapped to the [0, 1] range, and a predicted value is obtained. Predictions and probabilities can be mapped using the sigmoid function. The sigmoid function in equation (8) helps shrink the continuous input into a range of [0, 1]. Equation (11) $f(x)$ consists of features 'x_j' and their corresponding weights/coefficients 'β_j' in a linear form shown in equation (12).

$$\text{sig}(t) = \frac{1}{1 + e^{-t}}, \quad (10)$$

$$P(Y|X) = \frac{1}{1 + e^{-f(x)}}, \quad (11)$$

$$f(x) = x_0 + x_1\beta_1 + \cdots + x_k\beta_k + \varepsilon, \quad (12)$$

where $x, \beta, f(x) \in R^k$ and ε is the random error.

3.3.4. Stochastic Gradient Descent (SGD). This algorithm is good for huge training sets because this is a simplification of gradient descent algorithm. To reduce the computation cost, a stochastic version of the algorithm is being used in deep learning module training. Here, computing gradient is a more complex function, so the stochastic gradient will be descent trained on a random sample. The method is repeated over the training instances, updating the model parameters based on the update rule for each example as given in the following equation:

$$w \leftarrow w - \eta\left[\alpha\frac{\partial R(w)}{\partial w} + \frac{\partial L\left(w^T x_i + b, y_i\right)}{\partial w}\right], \quad (13)$$

where η is the learning rate, which controls the step size in the parameter space. The intercept 'b' is updated similarly but without regularization. The learning rate η can be either constant or gradually decaying. For classification, the default learning rate schedule is given in

$$\eta^{(t)} = \frac{1}{\alpha\left(t_0 + t\right)}, \quad (14)$$

where 't' is the time step, and t_0 is determined based on a heuristic.

3.3.5. Support Vector Classifier. SVM is a statistical learning theory algorithm for classification tasks that has different forms such as SVC. The SVC is used in pattern recognition problems and most commonly in document classification based on the statistical learning theory [15, 69]. SVM is the most widely used algorithm for the classification of text documents. SVCs learn n-dimensional hyperplanes to classify the linear and nonlinear data into appropriate classes. Let us consider a training set of labeled instances that are known as paired linear functions, and it can be expressed in the following equation:

$$y = \text{argmax}_{y'}\overrightarrow{\omega^T}\Phi\left(\overrightarrow{x}, y'\right), \quad (15)$$

$$\forall i \forall y = y_i\overrightarrow{\omega^T}\Phi\left(\overrightarrow{x_i}, y'\right) - \overrightarrow{\omega^T}\Phi\left(\overrightarrow{x_i}, y'\right) \ge 1 - \xi_i. \quad (16)$$

3.3.6. Linear Support Vector Classifier. One application of SVM classifiers is SVC, Linear SVC. They are working based on Library for SVMs (LIBSVM). Linear SVC works with more options, so it is flexible. SVC with parameter kernels equal to 'linear' has more flexibility in the choice of penalties and loss functions and should be scaled better to large numbers of samples.

3.4. Evaluation Metrics. The performance of different classification algorithms was analyzed with various FS and feature representation methods with different sizes of public benchmark Arabic text datasets. The proposed method performance is evaluated in terms of precision, recall, accuracy, and F-Measure as used subsequently [70–75].

3.4.1. Accuracy. In the measurement of a set, accuracy is defined as the closeness of the measured value to a specific actual value [76].

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{TN} + \text{FP}}. \quad (17)$$

3.4.2. Precision. Precision is the closeness among the measured quantities to each other. It is the fraction of retrieved items that are relevant to the classification results:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \quad (18)$$

3.4.3. Recall/Sensitivity. Recall in information retrieval is the fraction of the items successfully retrieved that are relevant to the posted query [46, 77]:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \quad (19)$$

(i) **Inputs**: $D = \{D_1, D_2, \ldots, D_n\}$
(ii) where n of Arabic documents
(iii) D_i selected document
(iv) Where $\forall (D_i) \exists C_j$ (C is name of class) and (j) number of classes
(v) **Output**: Assign D_i (unknown Document) to correct class Cj
(vi) **Begin** Read All collection of document in (corpus)
(vii) For $D = 1$ to n
(viii) Do Preprocessing for Document
(ix) $D[I]1 \leftarrow$ Tokenization ($D[i]$)
(x) $D[I]2 \leftarrow$ Stop word removal ($D[i]1$)
(xi) $D[I]3 \leftarrow$ Stemming($D[i]2$
(xii) $D[I] \leftarrow$ TFIDF($D[i]3$) OR $D[I] \leftarrow$ BoW($D[i]3$)
(xiii) $D[I]$Train = 80%
(xiv) $D[I]$Train = 20%
(xv) **"Training phase"** W = Input matrix with weights Train (TFIDF; Document)
(xvi) Weight(W) for document and add Label(L) for each document
(xvii) MNB $\leftarrow (W + L)$ where W is referred text and L referred to Label
(xviii) BNB $\leftarrow (W + L)$
(xix) SGD $\leftarrow (W + L)$
(xx) LR $\leftarrow (W + L)$
(xxi) SVC $\leftarrow (W + L)$
(xxii) Linear SVC $\leftarrow (W + L)$
(xxiii) **"Testing phase"** W = Input matrix with weights Train (BoW; Document)
(xxiv) MNB $\leftarrow (W)$ where W is referred text and L referred to Label
(xxv) BNB $\leftarrow (W)$
(xxvi) SGD $\leftarrow (W)$
(xxvii) LR $\leftarrow (W)$
(xxviii) SVC $\leftarrow (W)$
(xxix) MNB $\leftarrow (W)$
(xxx) **End for**
(xxxi) Push vector value without correspond label to classification algorithm then let the algorithm predict L
(xxxii) ML $\leftarrow$ (text)
(xxxiii) Class(L) $\leftarrow$ ML (Predict)
(xxxiv) End

Algorithm 1: Classification with TFIDF.

3.4.4. F-Measure. The F-measure is a measure of a test's accuracy. It considers both precision and the recall of the test samples to compute the score. The traditional F-measure or balanced F-score ($F1$ score) is the harmonic mean of precision and recall:

$$F1 - \text{score} = \frac{2 \cdot \text{TP}}{2 \cdot \text{TP} + \text{FP} + \text{FN}}. \tag{20}$$

3.5. Arabic Text Classification. One of the supervised machine learning techniques is a classification algorithm that is used to classify new observations on the testing based on learning the pattern from training data.

4. Datasets

Most of the researchers in Arabic text classification studies have collected their datasets individually. In this article, six well-known classifiers are operated on five publicly available Arabic datasets, namely, CNN, BBC, OSAC, ArCovidVac, and AlKhaleej Arabic corpora, and the performance metrics are studied [41, 78].

4.1. BBC Arabic Corpus. The Arabic BBC corpus dataset contains 4,763 documents in Arabic, and all documents belong to 1 class from 7. The number of documents for each class is as follows: 2,356 documents for Middle East News, 1,489 documents for world news, 296 documents for business and economics, 219 Sports documents, 49 documents for international press, 232 science and technology documents, and 122 art and culture documents [78]. In total, this corpus contains about 1,860,786 (1.8 M) words and 1,06,733 distinct keywords as summarized in Table 1.

4.2. CNN Arabic Corpus. The number of documents in the CNN Arabic corpus is 5,070. Every single document belongs to 1 class. The number of documents for each class is as follows: 836 business documents, 474 entertainment documents, 1,462 documents for middle east news, 526 documents for science and technology, 762 sports documents,

Table 1: BBC corpus dataset distribution.

#	Class type	No. of documents	Training/Validation set (80%)	Test set (20%)
1	Middle east news	2,356	1,885	471
2	Science and Technology	232	186	46
3	International press	49	39	10
4	Art and culture	122	98	24
5	Sports	219	175	44
6	Business and economy	296	237	59
7	World news	1,489	1,191	298

and 1010 documents for world news [78]. In total, this corpus contains about 2,241,348 (2.2 M) words and 1, 44, 460 distinct keywords, and the number of documents in each category is summarized in Table 2.

4.3. Open-Source Arabic Corpus (OSAc). This dataset included 22,429 Arabic text documents collected from multiple sources [78]. Every single text document belongs to 1 class from 10 classes. The number of documents for each class is different as mentioned in Table 3. This corpus contains about 1,81,83,511 (18M) words and 4,49,600 distinct keywords.

4.4. AlKhaleej. This dataset is scraped from all articles published in the news portal from 2008 to 2018 [41]. The collected text dataset exceeds the volume of 4 GB, and most of the articles are published on the websites. AlKhaleej has seven classes, which are Finance, Sports, Culture, Technology, Politics, Medical, and Religion. The dataset is generated in a balanced way where each class contains 6,500 Arabic articles. The training and testing sets are randomly split to be 5,200 and 1,300 per class, respectively.

4.5. Covid-19 Dataset. One more dataset about Covid-19 has been used to classify Arabic comments. Those comments belong to short text [79]. The data distribution of the Covid-19 dataset is shown in Table 4.

5. Experimental Analysis and Discussion

The algorithms such as MNB, BNB, LR, SGD Classifier, SVC, and linear SVC are implemented herein using Python 3.8.0 programming with Anaconda [Jupyter notebook]. The Python-based ML libraries such as NLTK, pandas, and scikit-learn are utilized to investigate the performance metrics by the proposed methods. The results and discussions concerning various techniques incorporated are highlighted in the subsequent sections.

5.1. Evaluation Result Based on Representation and Preprocessing. The six classification algorithms are operated on an unbalanced CNN benchmark dataset for ATC along with two types of representation such TF-IDF and BoW with and without preprocessing. Here, the SGD classifier and linear SVC with TF-IDF yield an accuracy of about 93%. In contrast, with BoW representation, LR followed by linear SVC offers good accuracy. For a balanced dataset (AlKhaleej), the linear SVC is found to be the best choice with preprocessing and in contrast, the accuracy decreases without preprocessing. Altogether, every classification algorithm except BNB works well and the best was when the dataset was preprocessed. The BNB algorithm works well only with binary classification, and its performance decreases with multiclass classification problems as listed in Tables 5 and 6. Another dataset has been studied, which is called Covid-19 as short text. We implemented the same idea to study the effectiveness of representation and preprocessing. We conclude that logistic regression and support vector classifiers are the best results and the effectiveness of preprocessing in some cases positive such as BoW and in others is negative such as TFIDF in Tables 7.

5.2. Evaluation Result Based on the Selected Features. From the experiments carried out for different numbers of features that have been chosen, we observed that as the number of features is increased, there is FDR drastic rise in the execution time. Meanwhile, no changes have been observed in the performance of the classifiers. It should be noted that the accuracy did not change with all the classifiers except SVC after increasing the features from 7,000 to 10,000. Another interesting observation is that the results are similar with the reduced and actual number of features, while SVC is operated on CNN. Conversely, while SVC is operated on the AlKhaleej dataset with 8,000 features, the accuracy was increased to 95% as listed in Figures 2 and 3.

5.3. Evaluation Result Based on Cross-Validation. The relationship between reduced or increased features with cross-validation is studied simultaneously with two representation methods, namely, TF-IDF and BoW. It is observed that an increased number of features would intensify or reduce the accuracy of ATC in a few cases. From the results summarized in Tables 8–15, it is inferred that the CNN with the increased number of features leads to performance improvement in Linear SVC. In contrast, there are no changes in the performance of Linear SVC that were observed with respect to increased features in the balanced dataset. The results prove that not only representation and feature selection affect the text classification performance, but also the nature of data (balanced/unbalanced) have a significant impact. With OSAC dataset, the accuracy remains the same among all the classifiers compared herein except BNB. The various interesting observations concerning the accuracy of ATC, while the features increased or decreased are highlighted in Tables 8–15.

Table 2: CNN corpus dataset distribution.

#	Class type	No. of documents	Train	Test
1	Entertainments	474	379	95
2	Science and technology	526	421	105
3	Sports	762	610	152
4	International press	-	-	-
5	Business and economy	836	669	167
6	World news	1,010	808	202
7	Middle east news	1,462	1,170	292

Table 3: OSAC corpus dataset distribution.

#	Class type	No. of documents	Train	Test
1	Health	2,296	1,837	459
2	Sports	2,419	1,935	484
3	Cooking recipes	2,373	1,898	475
4	Religion	3,171	2,537	634
5	Education and family	3,608	2,886	722
6	History	3,233	2,586	647
7	Economy	3,102	2,482	620
8	Stories	726	581	145
9	Low	944	755	189
10	Astronomy	557	446	111

Table 4: Covid-19 corpus dataset distribution.

#	Class type	No. of documents	Train	Test
1	Positive	7,962	6,369	1,592
2	Negative	635	508	136
3	Natural	1,391	1,113	286

Table 5: Data description of representation for CNN with and without pre-processing.

Classifiers	Bag of words (BoW)		Term frequency inverse document frequency (TFIDF)	
	With preprocessing	Without preprocessing	With preprocessing	Without preprocessing
Multinomial Naive Bayes	95	93	93	91
Bernoulli Naive Bayes	90	85	85	87
Logistic regression	97	96	96	94
Stochastic gradient descent	97	95	95	94
Support vector classifier	97	96	96	95
Linear support vector classifier	97	96	96	94

Table 6: Data description of representation for Alkhaleej with and without pre-processing.

Classifiers	Bag of words (BoW)		Term frequency inverse document frequency (TFIDF)	
	With preprocessing	Without preprocessing	With preprocessing	Without preprocessing
Multinomial Naive Bayes	88	88	64	58
Bernoulli Naive Bayes	61	73	61	73
Logistic regression	93	92	90	91
Stochastic gradient descent	91	91	93	92
Support vector classifier	90	91	90	92
Linear support vector classifier	92	91	93	92

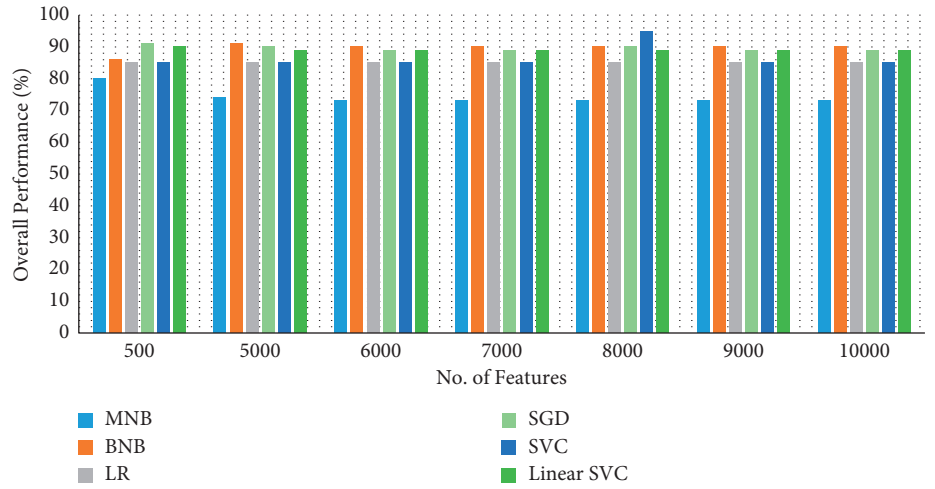

Figure 2: Data description of feature selection for CNN with TFIDF.

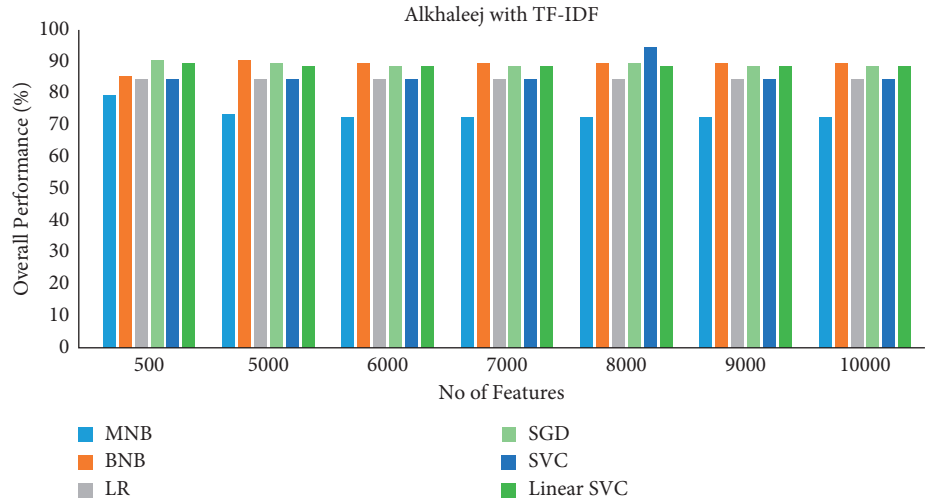

Figure 3: Data description of feature selection for Alkhaleej with TF-IDF.

Table 7: Data description of Covid-19 with BoW and cross-validation.

Classifiers	Bag of words (BoW)		Term frequency inverse document frequency (TFIDF)	
	With preprocessing	Without preprocessing	With preprocessing	Without preprocessing
Multinomial Naive bayes	80	81	80	80
Bernoulli Naive Bayes	80	80	75	73
Logistic regression	80	81	81	80
Stochastic gradient descent	79	80	80	80
Support vector classifier	80	80	81	80
Linear support vector classifier	79	79	80	80

Table 8: Data description of CNN with TIDF and cross-validation.

Classifiers	High accuracy with 500 features	Average accuracy for 10 cross-validation with 500 features
Multinomial Naive Bayes	90	88
Bernoulli Naive Bayes	81	78
Logistic regression	91	90
Stochastic gradient descent	92	90
Support vector classifier	93	91
Linear support vector classifier	94	91

Table 9: Data description of CNN with BoW and cross-validation.

Classifiers	High accuracy with 500 features	Average accuracy for 10 cross-validation with 500 features
Multinomial Naive Bayes	84	81
Bernoulli Naive Bayes	88	84
Logistic regression	94	91
Stochastic gradient descent	95	92
Support vector classifier	94	92
Linear support vector classifier	95	92

Table 10: Data description of BBC with TFIDF and cross-validation.

Classifiers	High accuracy with 500 features	Avg. Accuracy for 10 cross validation with 500 features
Multinomial Naive Bayes	84	80
Bernoulli Naive Bayes	88	86
Logistic regression	87	85
Stochastic gradient descent	93	91
Support vector classifier	87	85
Linear support vector classifier	93	90

Table 11: Data description of BBC with BoW and cross-validation.

Classifiers	High accuracy with 5000 features	Average accuracy for 10 cross-validation with 5000 features
Multinomial Naive Bayes	76	74
Bernoulli Naive Bayes	93	91
Logistic regression	88	85
Stochastic gradient descent	91	90
Support vector classifier	87	85
Linear support vector classifier	91	89

Table 12: Data description of OSAC with TFIDF and cross-validation.

Classifiers	High accuracy with 500 features	Average accuracy for 10 cross-validation with 500 features
Multinomial Naive Bayes	95	94
Bernoulli Naive Bayes	82	81
Logistic regression	97	97
Stochastic gradient descent	98	98
Support vector classifier	98	98
Linear support vector classifier	98	98

Table 13: Data description of OSAC with BOW and cross-validation.

Classifiers	High accuracy with 5000 features	Avg. Accuracy for 10 cross-validation with 5000 features
Multinomial Naive Bayes	95	95
Bernoulli Naive Bayes	87	86
Logistic regression	98	98
Stochastic gradient descent	99	99
Support vector classifier	98	98
Linear support vector classifier	99	99

Table 14: Data description of Alkhaleej 500 feature with TFIDF and cross-validation.

Classifiers	High accuracy with 500 features	Avg. Accuracy for 10 cross-validation with 500 features
Multinomial Naive Bayes	94	93
Bernoulli Naive Bayes	85	85
Logistic regression	96	96
Stochastic gradient descent	95	95
Support vector classifier	96	96
Linear support vector classifier	96	96

Table 15: Data description Alkhaleej 10000 feature with TFIDF and cross-validation.

Classifiers	High accuracy with 10000 features	Avg. Accuracy for 10 cross-validation with 10000 features
Multinomial Naive Bayes	95	95
Bernoulli Naive Bayes	90	90
Logistic regression	97	97
Stochastic gradient descent	97	97
Support vector classifier	97	97
Linear support vector classifier	97	97

6. Conclusion

For classifying Arabic text, a comprehensive investigation study is performed to show the effectiveness of preprocessing, feature extraction, feature selection, and the nature of the dataset. Several AI-based techniques have been presented to highlight the effectiveness of various methods on classifying Arabic text. The findings of the study show that there are many methods that affect the accuracy of the system performance on ATC. Our observation for this study proves that representation (feature extraction and feature selection) is highly important in ATC. At the same time, preprocessing, classification, and the nature of the dataset all affect the performance of classification. The results demonstrate the advantages of the feature representation approach that affect the text classification performance. Based on our understanding in this article, there are still many open issues for future work such as lack of benchmark dataset, lexicons, and simultaneously, finding techniques that handle the context meaning of ATC. Many other tools for ATC can be improved such as the augmentation of data. At last, an improving preprocessing technique for ATC specially is stemming.

Abbreviations

Arabic text classification:	ATC
Machine learning:	ML
Natural language processing:	NLP
Feature dimensionality reduction:	FDR
Vector space model:	VSM
Term class weight-inverse class frequency:	TCW-ICF
Term frequency-inverse document frequency:	TF-IDF
Out-of-vocabulary:	OOV
Information gain:	IG
Chi-square:	CHI
Mutual information:	MI
Information science research institute:	ISRI
Support vector classifier:	SVC
Linear support vector classifier:	LSVC
Multinomial Naive Bayes:	MNB
Bernoulli Naive Bayes:	BNB
Logistic regression:	LR
Stochastic gradient descent:	SGD
Principal component analysis:	PCA
Nonnegative matrix factorization:	NMF
Random projection:	RP
Linear discriminant analysis:	LDA.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

There are no conflicts of interest associated with publishing this paper.

Acknowledgments

This research was supported by the Researchers Supporting Project number (RSP-2021/244), King Saud University, Riyadh, Saudi Arabia.

References

[1] C. Aggarwal and C. C. Zhai, *Mining Text Data*, Springer, Berlin, Germany, 2012.

[2] M. Allahyari, S. Pouriyeh, M. Assefi et al., "A Brief Survey of Text Mining: Classification, Clustering and Extraction Techniques," 2017, https://arxiv.org/abs/1707.02919.

[3] M. suhail, C. Esteves, L. Sigal, and A. Makadia, "Mohammed Suhail Representation and Classification of Text Data, Ph.D. Thesis, University of Mysore, Des – 2019," p. 2019, 2019, https://arxiv.org/abs/2112.09687.

[4] R. Mamoun and M. Ahmed, "Arabic Text S temming: C omparative A nalysis," in *Proceedings of the 2016 Conf. Basic Sci. Eng. Stud. SGCAC 2016*, pp. 88–93, Khartoum, Sudan, February 2016.

[5] N. Y. Habash, *Introduction to Arabic Natural Language Processing*, vol. 3, no. 1, San Rafael, CA, USA, 2010.

[6] A. El Kah and I. Zeroual, "The effects of pre-processing techniques on Arabic text classification," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 10, no. 1, pp. 41–48, 2021.

[7] A. Hotho, "A Brief Survey of Text Mining," 2005.

[8] U. Naseem, I. Razzak, and P. W. Eklund, "A survey of preprocessing techniques to improve short-text quality: a case study on hate speech detection on twitter," *Multimedia Tools and Applications*, vol. 80, no. 28-29, pp. 35239–35266, 2020.

[9] M. Alhanjouri, "Pre processing techniques for Arabic documents clustering," *International Journal of Engineering Management*, vol. 7, no. 2, pp. 70–79, 2017, https://www.ijemr.net/DOC/PreProcessingTechniquesForArabicDocumentsClustering.PDF.

[10] M. O. Hegazi, Y. Al-Dossari, A. Al-Yahy, A. Al-Sumari, and A. Hilal, "Preprocessing Arabic text on social media," *Heliyon*, vol. 7, no. 2, Article ID e06191, 2021.

[11] Y. A. Alhaj, J. Xiang, D. Zhao, M. A. A. Al-Qaness, M. Abd Elaziz, and A. Dahou, "A study of the effects of stemming strategies on Arabic document classification," *IEEE Access*, vol. 7, pp. 32664–32671, 2019.

[12] L. Rigutini, "Automatic Text Processing: Machine Learning Techniques," pp. 1–146, 2004, Ph.D thesis.

[13] K. M. Hammouda and M. S. Kamel, "Efficient phrase-based document indexing for web document clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 10, pp. 1279–1296, 2004.

[14] G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," *Communications of the ACM*, vol. 18, no. 11, pp. 613–620, 1975.

[15] K. Kowsari, K. J. Jafari Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown, "Text classification algorithms: a survey," *Information*, vol. 10, no. 4, pp. 150–168, 2019.

[16] R. Al-Shalabi, G. Kanaan, and M. H. Gharaibeh, "Arabic text categorization using kNN algorithm," *Proc. 4th Int. Multiconference Comput. Sci. Inf. Technol.*vol. 15, no. 5–7, 2006.

[17] F. Sebastiani, "Machine learning in automated text categorization," *ACM Computing Surveys*, vol. 34, no. 1, pp. 1–47, 2002.

[18] Y. H. L. A. A. K. Jain, "Classification of text documents," *Computer Journal*, vol. 45, 1998.

[19] A. M. d. A. Mesleh, "Chi square feature extraction based SVMS Arabic language text categorization system," *Journal of Computer Science*, vol. 3, no. 6, pp. 430–435, 2007.

[20] F. Harrag and E. El-Qawasmah, "Neural network for Arabic text classification," in *Proceedings of the 2009 Second International Conference on the Applications of Digital Information and Web Technologies 2009*, pp. 778–783, London, UK, August 2009.

[21] A. M and K. Nigam, "A comparison of event models for naive Bayes text classi cation," *Environmental and Molecular Mutagenesis*, vol. 58, no. 8, pp. 582–591, 2017.

[22] S. Al-Harbi, A. Almuhareb, A. Al-Thubaity, M. S. Khorsheed, and A. Al-Rajeh, "Automatic Arabic text classification," *Text*, vol. 8, pp. 77–84, 2008, https://eprints.ecs.soton.ac.uk/22254/.

[23] S. A. Yousif, V. W. Samawi, I. Elkabani, and R. Zantout, "Enhancement of Arabic text classification using semantic relations of Arabic WordNet," *Journal of Computer Science*, vol. 11, no. 3, pp. 498–509, 2015.

[24] R. M. Duwairi, M. N. Al, and N. Khasawneh, "feature reduction techniques for Arabic text categorization," *Journal of the American Society for Information Science and Technology*, vol. 60, pp. 1852–1863, 2009.

[25] B. Al-Shargabi, W. Al-Romimah, and F. Olayah, "A comparative study for Arabic text classification algorithms based on stop words elimination," in *Proceedings of the 2011 International Conference on Intelligent Semantic Web-Services and Applications - ISWSA '11*, pp. 10–14, Amman, Jordan, April 2011.

[26] M. Hussein, "Improving Arabic text categorization using normalization and stemming techniques," *International Journal of Computers and Applications*, vol. 135, no. 2, pp. 38–43, 2016.

[27] S. M. Oraby, Y. El-Sonbaty, and M. Abou El-Nasr, "Exploring the Effects of Word Roots for Arabic Sentiment Analysis," in *Proceedings of the 6th International Joint Conference on Natural Language Processing*, pp. 471–479, Nagoya, Japan, October 2013.

[28] M. Mohd, "Effect of ISRI stemming on similarity measure for Arabic document clustering," *Information Retrieval Technology,Lecture Notes in Computer Science*, vol. 3699, 2011.

[29] A. Y. Muaad, H. J. Davanagere, M. A. Al-antari, and J. V. B. Benifa, "AI-based misogyny detection from Arabic levantine twitter tweets," in *Proceedings of the 1st Online Conf. Algorithms, 27 Sept. Oct. 2021*, pp. 4–11, MDPI Basel, Switzerland, October 2021.

[30] A. Kyriakopoulou and T. Kalamboukis, "Text classification using clustering," vol. 31, 2006.

[31] D. S. Guru, M. Ali, and M. Suhil, "A novel feature selection technique for text classification," *Advances in Intelligent Systems and Computing*, vol. 813, pp. 721–733, 2019.

[32] M. Al-Yahya, H. Al-Khalifa, A. Bahanshal, I. Al-Odah, and N. Al-Helwah, "An ontological model for representing semantic lexicons: an application on time nouns in the holy quran," *Arabian Journal for Science and Engineering*, vol. 35, no. 2 C, pp. 21–35, 2010.

[33] D. S. Guru, B. S. Harish, and S. Manjunath, "Symbolic representation of text documents," in *Proceedings of the Comput. 2010 - 3rd Proceedings of the Third Annual ACM Bangalore Conference on - COMPUTE '10*, pp. 1–4, Bangalore, India, January 2010.

[34] G. Sidorov, F. Velasquez, and E. Stamatatos, "Syntactic dependency-based N-grams as classification features," in *Proceedings of the 11th Mexican International Conference on Advances in Computational Intelligence*, pp. 1–11, San Luis Potos, Mexico, October 2012.

[35] X. Rong, "word2vec Parameter Learning Explained," pp. 1–21, 2014, http://arxiv.org/abs/1411.2738.

[36] C. D. M. Jeffrey-Pennington and R. Socher, "GloVe: Global vectors forWord representation," in *Proceedings of the 2014 Conf. Empir. Methods Nat. Lang. Process.*, Doha, Qatar, October 2014.

[37] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017.

[38] M. Peters, M. Neumann, M. Iyyer et al., "Deep contextualized word representations," *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, vol. 1, pp. 2227–2237, 2018.

[39] G. Kanaan, R. Al-Shalabi, S. Ghwanmeh, and H. Al-Ma'adeed, "A comparison of text-classification techniques applied to Arabic text," *Journal of the American Society for Information Science and Technology*, vol. 60, no. 9, pp. 1836–1844, 2009.

[40] M. Alhawarat and A. O. Aseeri, "A superior Arabic text categorization deep model (SATCDM)," *IEEE Access*, vol. 8, pp. 24653–24661, 2020.

[41] O. Einea, A. Elnagar, and R. Al Debsi, "SANAD: single-label Arabic news articles dataset for automatic text categorization," *Data in Brief*, vol. 25, Article ID 104076, 2019.

[42] F.-Z. El-Alami and S. O. El Alaoui, "Word sense representation based-method for Arabic text categorization," in *Proceedings of the 2018 9th International Symposium on Signal, Image, Video and Communications (ISIVC)*, pp. 141–146, Rabat, Morocco, November 2018.

[43] F.-Z. El-Alami, S. O. El Alaoui, and N. En-Nahnahi, "Deep neural models and retrofitting for Arabic text categorization," *International Journal of Intelligent Information Technologies*, vol. 16, no. 2, pp. 74–86, 2020.

[44] E. Mahdaouy and E. Alaoui, "A deep autoencoder-based representation for Arabic text categorization," *Journal of Information and Communication Technology*, vol. 19, no. 3, pp. 381–398, 2020.

[45] A. Y. Muaad, H. Jayappa, M. A. Al-antari, and S. Lee, "ArCAR: a novel deep learning computer-aided recognition for character-level Arabic text representation and recognition," *Algorithms*, vol. 14, no. 7, p. 216, 2021.

[46] A. Y. Muaad, M. A. Al-antari, S. Lee, and H. J. Davanagere, "A novel deep learning ArCAR system for Arabic text recognition with character-level representation," *Ioca 2021*, vol. 14, no. 7, pp. 1–7, 2021.

[47] A. Nehar, D. Ziadi, and H. Cherroun, "Rational kernels for Arabic root extraction and text classification," *Journal of King Saud University - Computer and Information Sciences*, vol. 28, no. 2, pp. 157–169, 2016.

[48] S. Bahassine, A. Madani, M. Al-Sarem, and M. Kissi, "Feature selection using an improved Chi-square for Arabic text classification," *Journal of King Saud University - Computer and Information Sciences*, vol. 32, no. 2, pp. 225–231, 2020.

[49] Y. A. Alhaj and M. A. A. Al-qaness, "Feature selection on Arabic document classification: comparative study feature selection on Arabic document classification: comparative study," in *Proceedings of the 15 ICIM*, pp. 345–355, Yamaguchi University, Japan, April 2018.

[50] A. A. Mohamed, "An effective dimension reduction algorithm for clustering Arabic text," *Egyptian Informatics Journal*, vol. 21, no. 1, pp. 1–5, 2020.

[51] H. Chantar, M. Mafarja, H. Alsawalqah, A. A. Heidari, I. Aljarah, and H. Faris, "Feature selection using binary grey wolf optimizer with elite-based crossover for Arabic text classification," *Neural Computing & Applications*, vol. 32, no. 16, pp. 12201–12220, 2020.

[52] W. Alabbas, H. M. Al-Khateeb, and A. Mansour, "Arabic text classification methods: systematic literature review of primary studies," in *Proceedings of the 2016 4th IEEE International Colloquium on Information Science and Technology (CiSt)*, pp. 361–367, Tangier, Morocco, October 2016.

[53] M. El Kourdi, A. Bensaid, and T. E. Rachidi, "Automatic Arabic Document Categorization Based on the Naïve Bayes Algorithm," in *Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages-Semitic '04*, Geneva, Switzerland, August 2004.

[54] M. Saad, "The impact of text preprocessing and term weighting on Arabic text classification," *Arabic text mining*, p. 112, 2010, https://site.iugaza.edu.ps/msaad/files/2012/05/mksaad-Arabic-text-classification-MSc-Thesis-2010-rev9.pdf.

[55] Z. Jianqiang and G. Xiaolin, "Comparison research on text pre-processing methods on twitter sentiment analysis," *IEEE Access*, vol. 5, pp. 2870–2879, 2017.

[56] Z. Alyafeai, M. S. Al-shaibani, M. Ghaleb, and I. Ahmad, "Evaluating various tokenizers for Arabic text classification," vol. 5, 2021, https://arxiv.org/abs/2106.07540.

[57] E. T. Al-shammari, J. Lin, and D. Ph, "Towards an Error-free Arabic Stemming," in *Proceedings of the 2nd ACM workshop on Improving non english web searching*, pp. 9–15, Napa Valley, CL, USA, October 2008.

[58] Y. A. Alhaj, W. U. Wickramaarachchi, A. Hussain, M. A. A. Al-Qaness, and H. M. Abdelaal, "Efficient feature representation based on the effect of words frequency for Arabic documents classification," in *Proceedings of the 2nd International Conference on Telecommunications and Communication Engineering-ICTCE 2018*, pp. 397–401, Beijing, China, November 2018.

[59] M. Elhag, M. Abo, N. Idris, R. Mahmud, and A. Qazi, "A multi-criteria approach for Arabic dialect sentiment analysis for online reviews: exploiting optimal machine learning algorithm selection," *Sustainability*, vol. 13, no. 18, Article ID 10018, 2021.

[60] O. Bousquet and L. Bottou, "The tradeoffs of large scale learning," *Advances in Neural Information Processing Systems*, vol. 20, pp. 161–168, 2007.

[61] L. Jiang, S. Wang, C. Li, and L. Zhang, "Structure extended multinomial naive Bayes," *Information Sciences*, vol. 329, pp. 346–356, 2016.

[62] S. Ruan, H. Li, C. Li, and K. Song, "Class-specific deep feature weighting for naïve Bayes text classifiers," *IEEE Access*, vol. 8, pp. 20151–20159, 2020.

[63] A. M. Khonsa Izzaty, M. S. Mubarok, N. S. Huda, and Adiwijaya, "A multi-label classification on topics of quranic verses in English translation using tree augmented naïve Bayes," in *Proceedings of the 2018 6th International Conference on Information and Communication Technology (ICoICT)*, no. 1, pp. 103–106, Bandung, Indonesia, May 2018.

[64] G. Singh, B. Kumar, L. Gaur, and A. Tyagi, "Comparison between multinomial and Bernoulli naïve Bayes for text classification," in *Proceedings of the 2019 International Conference on Automation, Computational and Technology Management (ICACTM) 2019*, pp. 593–596, London, UK, April 2019.

[65] R. Basis, F. Classifier, and S. Mapping, "A comparative study of kernel logistic regression, radial basis function classifier, multinomial naïve Bayes, and logistic model tree for flash flood susceptibility mapping," *Water*, vol. 12, no. 1, p. 239, 2020.

[66] K. Sarkar and M. Bhowmick, "Sentiment polarity detection in Bengali tweets using multinomial Naïve Bayes and support vector machines," in *Proceedings of the 2017 IEEE Calcutta Conference (CALCON)*, pp. 31–35, Kolkata, India, January 2018.

[67] H. S. Ibrahim Kaibi, "Sentiment analysis approach based on combination ofWord embedding techniques," *Embedded Systems and Artificial Intelligence,Advances in Intelligent Systems and Computing*, vol. 1171, 2021.

[68] A. Y. Muaad, H. J. Davanagere, J. V. Bibal Benifa et al., "Artificial intelligence-based approach for misogyny and sarcasm detection from Arabic texts," *Computational Intelligence and Neuroscience*, vol. 2022, Article ID 7937667, 9 pages, 2022.

[69] I. L. Us, B. Ch, and B. Zhang, "Clustering Based Text Classification," in *Proceedings of the Third IEEE International Conference on Data Mining*, Melbourne, FL, USA, November 2008.

[70] C. Chola, M. B. B. Heyat, F. Akhtar et al., "IoT based intelligent computer-aided diagnosis and decision making system for health care," in *Proceedings of the 2021 International Conference on Information Technology (ICIT)*, pp. 184–189, Amman, Jordan, July 2021.

[71] C. Chola, J. V. B. Benifa, D. S. Guru et al., "Gender identification and classification of *Drosophila melanogaster* flies using machine learning techniques," *Computational and Mathematical Methods in Medicine*, vol. 2022, Article ID 4593330, 9 pages, 2022.

[72] M. A. Al-antari, M. A. Al-masni, M.-T. Choi, S.-M. Han, and T.-S. Kim, "A fully integrated computer-aided diagnosis system for digital X-ray mammograms via deep learning detection, segmentation, and classification," *International Journal of Medical Informatics*, vol. 117, pp. 44–54, 2018.

[73] M. A. Al-antari, S.-M. Han, and T.-S. Kim, "Evaluation of deep learning detection and classification towards computer-aided diagnosis of breast lesions in digital X-ray mammograms," *Computer Methods and Programs in Biomedicine*, vol. 196, Article ID 105584, 2020.

[74] M. A. Al-antari, C.-H. Hua, J. Bang, and S. Lee, ""Fast deep learning computer-aided diagnosis of COVID-19 based on

digital chest x-ray images",," *Applied Intelligence*, vol. 51, no. 5, pp. 2890–2907, 2021.

[75] M. A. Al-masni, M. A. Al-antari, J.-M. Park et al., "Simultaneous detection and classification of breast masses in digital mammograms via a deep learning YOLO-based CAD system," *Computer Methods and Programs in Biomedicine*, vol. 157, pp. 85–94, 2018.

[76] J. Hanumanthappa, A. Y. Muaad, J. V. Bibal Benifa, C. Chola, V. Hiremath, and M. Pramodha, "IoT-Based Smart Diagnosis System for HealthCare," *Healthcare Monitoring and Data Analysis using IoT: Technologies and applications*, vol. 93, pp. 461–469, 2022.

[77] M. Pramodha, A. Y. Muaad, J. Hanumanthappa, C. Chola, and A. Mugahed, "A hybrid deep learning approach for COVID-19 diagnosis via CT and X - R ay medical images," vol. 2, pp. 1–10, 2021.

[78] M. Saad and W. Ashour, "OSAC: open source Arabic corpora," in *Proceedings of the 6th International Conference Electrical Engineering and Computer Systems and Science (EECS'10)*, pp. 118–123, Lefke, North Cyprus., November 2010.

[79] H. Mubarak, S. Hassan, S. A. Chowdhury, and F. Alam, "ArCovidVac: Analyzing Arabic Tweets about COVID-19 Vaccination," 2022, http://arxiv.org/abs/2201.06496.