Hindawi
Scientific Programming
Volume 2021, Article ID 1367210, 9 pages
https://doi.org/10.1155/2021/1367210

Research Article

The Use of Stemming in the Arabic Text and Its Impact on the Accuracy of Classification

Jaffar Atwan,[1] **Mohammad Wedyan**,[2] **Qusay Bsoul**,[3] **Ahmad Hammadeen**,[4] **and Ryan Alturki**[5]

[1] *Department of Computer Information Systems, Al-Balqa Applied University, Al-Salt, Jordan*
[2] *Faculty of Artificial Intelligence, Al-Balqa Applied University, Al-Salt, Jordan*
[3] *Cybersecurity Department, Science and IT College, Irbid National University, Irbid, Jordan*
[4] *Department of Computer Science, Al-Balqa Applied University, Al-Salt, Jordan*
[5] *Department of Information Science, College of Computer and Information Systems, Umm Al-Qura University, Makkah, Saudi Arabia*

Correspondence should be addressed to Mohammad Wedyan; mwedyan@bau.edu.jo

Received 13 August 2021; Revised 12 October 2021; Accepted 21 October 2021; Published 24 November 2021

Academic Editor: Muhammad Usman

Copyright © 2021 Jaffar Atwan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The ongoing growth in the vast amount of digital documents and other data in the Arabic language available online has increased the need for classification methods that can deal with the complex nature of such data. The classification of Arabic plays a large and important role in many modern applications and interferes with other sciences, which start from search engines and do not end with the Internet of Things. However, addressing the Arab classification errors with high performance is largely insufficient to deal with the huge quantities to reveal the classification of Arab documents; while some work was tackled out on the classification of the Arabic text, most of the research has focused on English text. The methods proposed for English are not suitable for Arabic as the morphology of the two languages differs substantially. Moreover, morphologically, the preprocessing of Arabic text is a particularly challenging task. In this study, three commonly used classification algorithms, namely, the K-nearest neighbor, Naïve Bayes, and decision tree, were implemented for Arabic text in order to assess their effectiveness with and without the use of a light stemmer in the preprocessing phase. In the experiment, a dataset from Agency France Persse (AFP) Arabic Newswire 2001 consisting of four categories and 800 files was classified using the three classifiers. The result showed that the decision tree with light stemmer had the best accuracy rate for classification algorithm with 93%.

1. Introduction

Machine learning (ML) is a branch of Artificial Intelligence (AI) research [1], which aims to develop practically relevant multipurpose algorithms based on a little amount of data. Difference between ML approaches and general AI lies on the discovered patterns in data and the way in which data is used. There are different examples of the application of ML, such as fraud discovery, weather forecasting, and patients' diagnosis. The two major forms of ML are supervised and unsupervised learning. Here we consider the former, which involves the generation of a mapping from labeled training data into an output of predictions or classes. This process can be described as classification and is the core aspect of supervised ML.

Classification involves the determination of output values known as classes or labels using input objects. This mapping is known as a model or classifier. The entered objects are related to the categorized objects also recognized as examples, instances, or tuples. According to [2], ML classification technique involves combining several instances together with their known labels by manually tagging a group of instances. The group of labeled instances is recognized as a training set. The labeled instances (i.e., training set) are used by classifier to generate the model that maps the instance to its label. As a result, then the training model can be used to label or classify new, unknown instances. In the current study, which focuses on the classification of Arabic text, the instances are carefully chosen from a prelabeled pool of instances by employing enhanced Arabic classifiers.

There are many situations in which unlabeled documents are both plentiful and cheap. However, labeling them is regarded as costly and time-consuming. For example, it is handy to get a huge amount of documents basically with no price; in contrast a lot of money is paid for human comment hosts to classify these documents with their subject classification whether they are in Arabic or non-Arabic. Also, data for videos is easy to collect, but it is very difficult to get good semantic content labels from that data. Likewise, it is easy to get a wide range of compounds that may be useful for treating a disease, but it is very expensive to run expensive biochemical tests to see which one really works. These three examples are essentially classification problems.

Several algorithms have been implemented to solve the text classification (TC) problem. More than one work in this field has focused on English text. In contrast, little research has been done on the Arabic script. The English text differs from the Arabic text in terms of its morphological structure, which makes the preprocessing of Arabic text more challenging for a number of reasons. The aim of the study is to evaluate the performance of the Arabic text classification system using three distinct categorization methods, namely, the decision tree (DT), Naïve Bayes (parametric-based), and K-nearest neighbor (KNN) (example-based) classifiers. In order to get the best integration of weighting scheme and technique, various weighting schemes were adopted in the first two methods.

In the following, Section 2 discusses text classification. Then we present the motivation and objective of this work in Section 3. An overview of the related works and the three classifiers considered in this study are provided in Section 4. Section 5 introduces the framework of the proposed Arabic text classifier. Section 6 describes the experiment and Section 7 presents the document representation. Section 8 presents results and Section 9 contains the conclusion and details of future work.

2. Text Classification

Text classification is a machine learning supervised task requiring prelabeled documents in need of learning. Furthermore, it aims to detect new documents based on certain learned criteria [3]. Applications of text-based knowledge and the TC feature are particularly important in natural language processing (NLP), at least because of the recent increase in the volume of available text data. One example of an area in which TC and NLP are needed is filtering [4], which is a process that attempts to filter a user's inbound documents to identify those that are unwanted or unsolicited. Another is sentiment analysis [5], which looks to identify the general feelings cleared up in a document in order to measure, for example, customer satisfaction.

It is possible to apply the supervised learning algorithms of the classification training model to a set of respective problem states to overcome the problems encountered in the TC. These models can then be used to identify the unlabeled document class [2, 6–10].

There are two phases in the TC approach: training and testing. The training phase involves the building of a classifier using a group of the collected documents (called the training set) and by allocating a subset of the training set to each category before processing them via several NLP techniques. The aim of this processing is to extract the set of features from the training set which will be used as the representative for each category. The remainder of the collected documents is the so-called test set, which is used in the testing stage to evaluate the performance of the classifier in terms of its ability to classify the documents that it has not seen before into the correct categories, where performance is assessed by comparing the categories selected by the classifier with those of the predefined documents [3].

A TC system generally consists of these parts:

(i) Text preprocessing, which converts the text into a group of dimensions that can be processed by classifiers.

(ii) Reducing dimensionality, which decreases features number to enhance the efficiency of classification algorithms. This can be done using methods such as feature selection and dimension reduction [8, 9, 11, 12].

(iii) Classifier training, which is the process of building an autonomous classifier using supervised learning frameworks [2].

(iv) Prediction, which is the process of using a trained classifier to generate labels for new documents [2].

It has been indicated in [13] that texts can be symbolically represented as a set of characteristics by employing two representation methods, namely, the n-gram and the bag of words (BOW). The former involves the use of some words or sentences as characteristics while the latter employs the order of the words or characters of *n* length. Past studies [14, 15] have pointed out that the creation of an accurate TC system requires the effective handling of a high number of characteristics or features (which may be number in their tens of thousands). Hence some information retrieval (IR) techniques such as stemming and elimination of stop-words have been used to decrease the feature space dimensionality.

3. Motivation and Objectives

The importance of using technologies for classification has increased due to the need to have the ability to automatically classify the huge amounts of diverse text-based information that can be found on the Internet and in electronic/digital format in many languages, including Arabic. Hence, several studies initially focused on addressing the challenges associated with standard Arabic document classifiers [6, 7, 9, 16], which then encouraged more studies that concentrated on enhancing the performance of Arabic document classifiers. This research continues because most Arab classifiers are characterized by their inability to deal accurately with the vast quantities of documents that have been identified as Arabic documents. As such, this is considered the major problem in the classification of Arabic texts.

One of the main obstacles facing researchers working in the field of text classification for documents in Arabic is the failure of the available classifiers to deal with stemming, which is a factor that might affect other processes in a document classification system. To address this issue, an algorithm is employed to define the stemming rule, and this rule depends on the processing of grammatical components of an utterance to solve the complexity of morphological and syntax.

The major TC problem is related with the enormous features extracted from the text (can reach hundreds or thousands). Therefore, the time required to substitute a term with its possible concepts may increase and the high dimensionality of the feature space may reduce classifier performance. The number of features or feature size can be reduced by extracting the essential semantics from texts [17, 18].

Therefore, in order to reduce the feature size of Arabic text, this study evaluates three classifiers without and with stemming [19]. It is hoped that the outcome of this research will contribute to the improved tracking and detection of new documents and their categorization into the relevant categories and consequently, the improved performance of Arabic classifiers. In sum, this study attempts to answer the following research question: What is the effect of classification techniques on Arabic documents without or with the use of stemmer?

4. Related Works

Text classification refers to assigning predefined categories of text depending on the content of the documents. For natural language processing and other applications of textual knowledge, text classification is important. The importance of text classification is due to the recent increase in the volume of available text data. It is possible to overcome the problems of text classification by applying supervised learning algorithms to train the models of classification with a group of abovementioned examples of the problem in question that clarify correct classification (labels). These models can then be used to predict the labels of unlabeled documents [12, 20–23]. A text classification system may be built from the following components.

It is supposed that the structure of categories is known in advance in the case of supervised algorithms, and these algorithms require a group of tagged documents to map the documents to some prespecified classes. However, as abovementioned, in case of huge dataset it is difficult to remark the true label and class of the document in training set. Hence, the focus and review in this section will be on the most commonly used classification based on algorithms, namely, KNN, NB, and DT.

4.1. K-Nearest Neighbor Algorithm (KNN) Classifier. KNN is a popular example-based classifier. There are two basic steps, the KNN was developed as a popular instance-based learning technique which has been efficient in several text categorization tasks. The flow of the algorithm is boiled down as follows: first, the k-nearest neighbors are found within the given training documents [24]. Second, the test document category is found using the category labels of these neighbors. The conventional approach usually assigns the test document with the commonest label of category among the established k-nearest neighbors.

The conventional KNN is the basis of the extended weighted kNN in which the contribution of each neighbor is weighted with respect to its proximity to the test document. Next, the similarity of the adjacent documents in each class is collected to obtain the document class score; i.e., the class score cj for x document is illustrated as follows:

$$Score\ (cj, x)\ =\ \sum (di\ \ \in N(x))\cos(x,\ di).y(di\ ,cj), \tag{1}$$

where the training document is $= di$, group of x nearest k training document is $= N(x)$, cos $(x, di) =$ the cosine similarity between x and d_i, and $y\ (di\ , cj\ ) = a$ function with a value of 1 if d_i is relevant to class cj, and 0 else. The class with the highest score allocates x test document.

4.2. Naïve Bayes (NB) Classifier. The NB classifier is a simple probabilistic-based classifier, which is based on Bayes' theorem which estimates the likelihood of the classes assigned to a test document using the joint probabilities of terms and classes of such document. The naïve aspect of the classifier originated from its assumption of the conditional independence of all terms of each category from the other category. Based on this assumption of independence, the parameters of each term can be separately learned, as such, making the computation operations easier compared to the non-NB classifiers. An NB proper classifier can merely assume that there is no relation between the presence or the nonappearance of a particular category trait with any other feature. We can express this presumption as follows:

$$P(C\_i|d) = \frac{(P(C\_i)P(d|C\_i))}{(P(d))}, \tag{2}$$

where $P(C_i\ |d)$ refers to the previous probability of class C_i in the presence of a new instance d and $P(C_i)$ symbolizes toe probability of class C_i, which can be figured by

$$P(\mathrm{Ci}) = \frac{\mathrm{Ni}}{\mathrm{N}}, \tag{3}$$

where the proper samples that are associated with class $C_i = N_i$, N is the number of classes, the likelihood of a sample d being assigned to a class $C_i = P(d|C_i)$, and the likelihood of sample $d = P(\mathrm{d})$.

4.3. Decision Tree (DT) Classifier. The DT is a commonly used inductive learning method that is characterized by its ability to resist noisy data and its ability to learn detailed expressions, which makes it suitable for document classification [25]. This algorithm employs a "divide and conquer" approach, where it divides complex decisions into several simpler ones.

It divides complex decisions into several simpler ones. In the learning stage of the DT, it is contained from a group of tagged training examples manifested in a record of features values and a label class due to big areas of decision tree learning and search are top-down, repeated process and greedy start with an empty tree and the entire training data. A feature has more information about content and has a best partition chosen as the splitting feature for the training data and for the root and then the training data is divided into disjoint subgroups satisfying the values of the incision features. In respect of every subgroup, the algorithm occurs before repeatedly until each subgroup's classes maintain the same class [3].

5. Framework of Arabic Text Classifier

When answering the user's demand, the TC system requests to get the following: classify the intended document, classify it swiftly, meet user requirements, and obtain optimum classification efficacy [26, 27]. Thus, the objective of the Arabic TC (ATC) structure presented in this study is to raise the ATC system efficiency, if the system takes into account the semantic relationship and the complexity of the Arabic terms.

The ATC framework depends on the following stages: preprocessing, extraction, representation, application of classifiers, and evaluation. The ATC framework takes into account these important issues (Figure 1).

The ATC system's first step is the preprocessing phase, which is an important step for document presentation. It involves the initial processing of the text to choose the appropriate terms to be indexed. Through the preprocessing phase, many operations are performed like stemming, stop-words eliminations, tokenization, and normalization.

In this study, the main contribution is to build an automatically Arabic text classifier to classify documents based on morphological knowledge representation by utilizing a light stemmer. The general procedures performed in this method are as follows (Figure 2).

Figure 3 shows the different stages of the ATC framework which will be discussed in detail in Section 6, "Experiment."

6. Experiment

Arabic-language classification is a supervised learning-dependent process; three ML processes and supervised algorithms were used in this experiment, the KNN, NB, and DT classifiers [28]. In order to enhance the accuracy of the Arabic classifier, the Arabic Light10 stemmer was employed and tested. In this section, the steps shown earlier in the Arabic text classifier framework were presented and tested.

6.1. Dataset. We used a dataset that consisted of 800 documents that were classified into four classes. These documents were extracted from the relevant documents for four queries (i.e., each query represents class) from an Arabic Newswire dataset that were used recently in TREC experiments [29]. Figure 4 shows a sample document from the dataset.

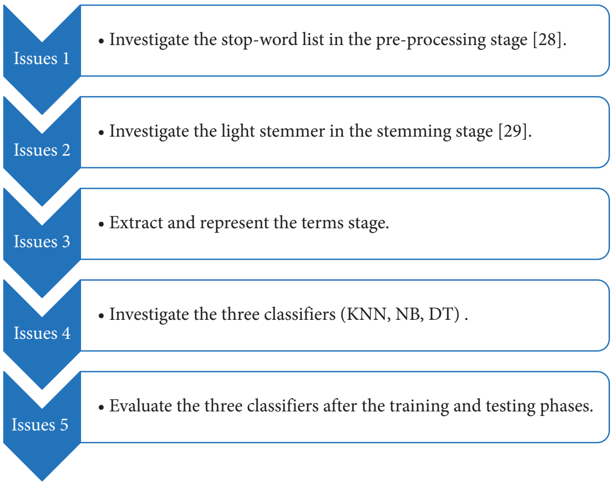

Issues 1
• Investigate the stop-word list in the pre-processing stage [28].
Issues 2
• Investigate the light stemmer in the stemming stage [29].
Issues 3
• Extract and represent the terms stage.
Issues 4
• Investigate the three classifiers (KNN, NB, DT) .
Issues 5
• Evaluate the three classifiers after the training and testing phases.

Figure 1: Important issues.

6.2. Preprocessing. The aim of the preprocessing phase is to filter out nonsignificant data, such as tags (i.e., <DOC>, <DOCNO >, <DOCTYPE>, <DATE_TIME>, <BODY>, <TEXT>, <END_TIME>) from a document. In carrying out the step of preprocessing, the document must be converted into a format suitable for the representation process so that learning algorithms are applied. Following this, removal of the unnecessary words used as the characters such as punctuation and special markers takes place. Thus, in carrying out this step, three commonly identified tasks, tokenization and normalization, stop-word removal (in order to reduce the dimension of the feature space), and mainly stemming and lemmatization, need to be done. Based on the review of these tasks in previous studies, the following section provides a brief description of these three tasks.

6.3. Tokenization and Normalization of Data. According to [31], text documents are usually converted in a way that is appropriate for their analysis by employing a machine learning algorithm. The text is divided into separate units by using either spaces or special symbols. As such, every word in a text is represented as a single unit. This procedure is called tokenization. For instance, (خير جليس في الزمان كتاب) it can be tokenized using white space to list of tokens (words) as (خير، جليس، في، الزمان، كتاب). Accordingly, the other task known as normalization is useful because this is done before the task stemming particularly for the Arabic script. This is for the reason that the text normalization in the Arabic language helps in the downgrading the various shapes of characters to produce a uniformed shape representing these shapes. This is illustrated by the following example:

(i) Substitute آ, إ as well as أ by ا

(ii) Substitute the last ة by ه

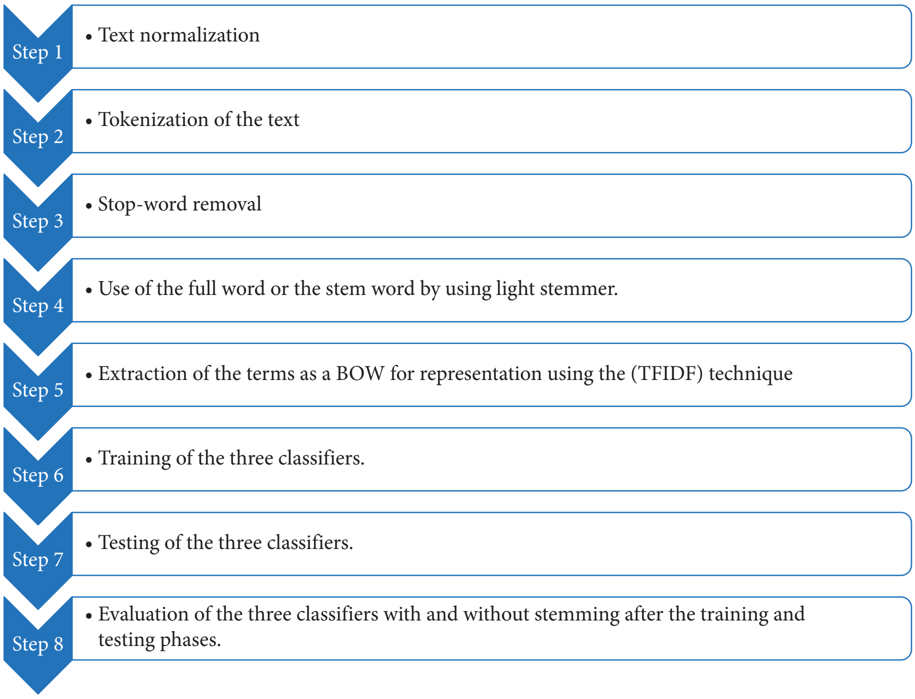

Step 1
• Text normalization
Step 2
• Tokenization of the text
Step 3
• Stop-word removal
Step 4
• Use of the full word or the stem word by using light stemmer.
Step 5
• Extraction of the terms as a BOW for representation using the (TFIDF) technique
Step 6
• Training of the three classifiers.
Step 7
• Testing of the three classifiers.
Step 8
• Evaluation of the three classifiers with and without stemming after the training and testing phases.

Figure 2: General procedures performed in this method.

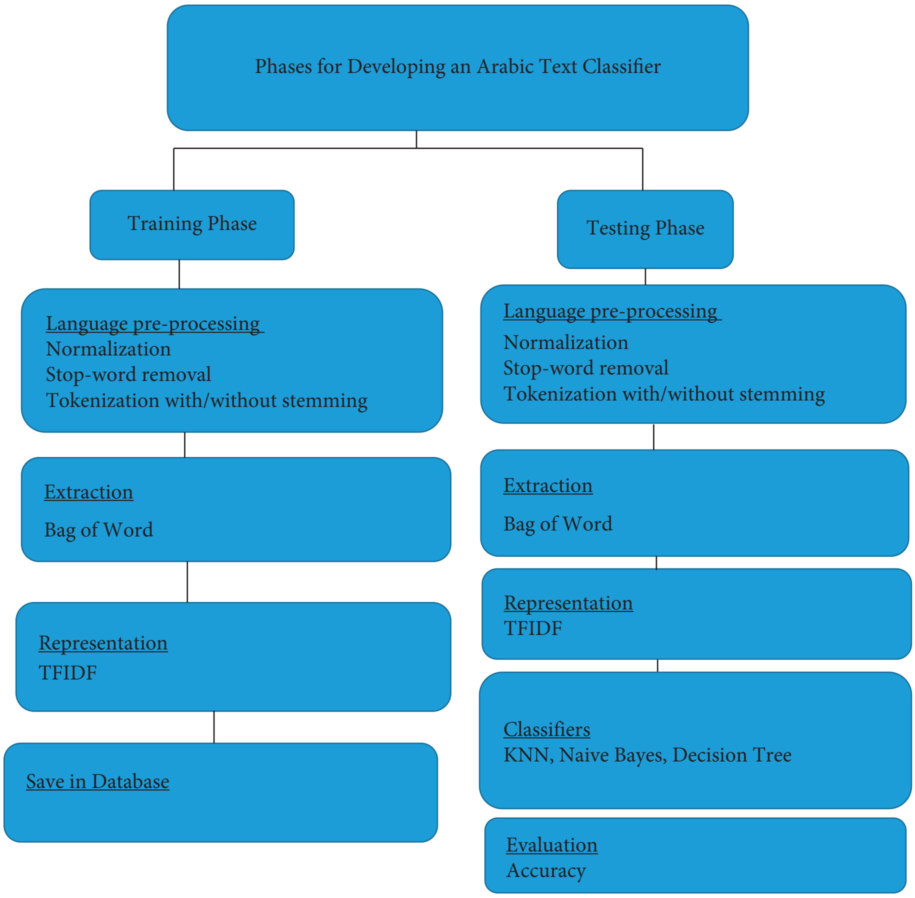
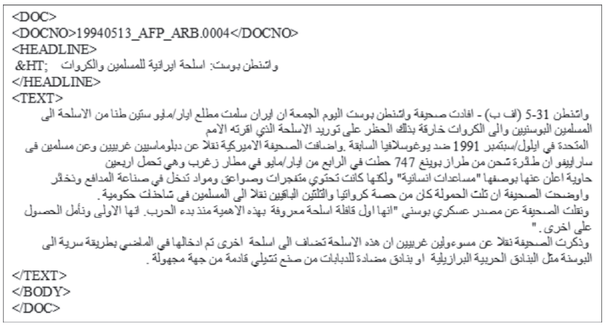


Figure 3: Phases for developing an Arabic classifier.



```
<DOC>
<DOCNO>19940513_AFP_ARB.0004</DOCNO>
<HEADLINE>
 &HT;   واشنطن بوست: اسلحة ايرانية للمسلمين والكروات
</HEADLINE>
<TEXT>
 واشنطن 5-31 (اف ب) - افادت صحيفة واشنطن بوست اليوم الجمعة ان ايران سلمت مطلع ايار/مايو ستين طنا من الاسلحة الى
المسلمين البوسنيين والى الكروات خارقة بذلك الحظر على توريد الاسلحة الذي اقرته الامم
 المتحدة في ايلول/سبتمبر 1991 ضد يوغوسلافيا السابقة .واضافت الصحيفة الاميركية نقلا عن دبلوماسيين غربيين وعن مسلمين فى
ساراييفو ان طائرة شحن من طراز بوينغ 747 حطت في الرابع من ايار/مايو في مطار زغرب وهي تحمل اربعين
حاوية اعلن عنها بوصفها "مساعدات انسانية" ولكنها كانت تحتوي متفجرات وصواعق ومواد تدخل في صناعة المدافع وذخائر
 واوضحت الصحيفة ان ثلث الحمولة كان من حصة كرواتيا والثلثين الباقيين نقلا الى المسلمين فى شاحنات حكومية .
 ونقلت الصحيفة عن مصدر عسكري بوسني "انها اول قافلة اسلحة معروفة بهذه الاهمية منذ بدء الحرب. انها الاولى ونأمل الحصول
على اخرى ."
 وذكرت الصحيفة نقلا عن مسوءولين غربيين ان هذه الاسلحة تضاف الى اسلحة  اخرى تم ادخالها في الماضي بطريقة سرية الى
البوسنة مثل البنادق الحربية البرازيلية  او بنادق مضادة للدبابات من صنع تشيلي قادمة من جهة مجهولة .
</TEXT>
</BODY>
</DOC>
```

Figure 4: Sample document from TREC 2001 collection [30].

(iii) Substitute the last ى by ي

6.4. *Elimination of Stop-Words.* Stop-words are those words that occur frequently in the document. These words give no hint to the document content in which they appear. Stop-words removal is mandatory prior to submitting text to be processed by an ATC system in order to reduce both time and cost. Hence a list of stop-words is created, which is then applied to the indexed terms to be eliminated. However, for an ATC system there is no prominent stop-words list that could be used in such systems. Consequently, for the experiment, the same stop-words list used in [32] was used here. Table 1 provides some examples of Arabic stop-words.

6.5. *Stemming Text.* The text stemming process helps in reducing the various inflectional derivational words forms to a uniform called the stem [32]. For instance, the terms, "work," "works," "working," "worked," and "worker" are derived from the "work" stem. Table 2 shows an example of different Arabic words derived from the same root. The word root is gained by eliminating some or all the word suffixes attached to it. In the ATC system, terms are grouped together that share the same stem or root, which effectively raises the number of matched documents to the user query. Furthermore, there is an overall improvement in the ATC performance due to the reduction in the dictionary size as a result of the stemming process [33].

In this paper, for stemming purpose we followed the same stemming steps in [33] using Light10 stemmer, as follows:

(1) Remove "و" ("and") for Light2, Light3, Light8, and Light10 if the remainder of the word is three or more characters long

(2) Eliminate the definite articles that leave the remaining word with more than or equal to two letters

(3) Keep words with a length of two or more letters after suffixes removal which appears in the list; remove one at a time in order from right to left

Table 3 shows the list of strings that should be removed. Note that the conjunction and definite articles are the prefixes shown in the table. No elimination is done for the strings that deemed an actual Arabic prefix in Light10 stemmer.

Table 1: Examples of Arabic stop-words.

Arabic word	English meaning
في	In
من	From
إلى	To
على	Over
عن	About

Table 4 shows an example of affixes in Arabic word.

7. Document Representation

Each document in the study dataset was represented by a vector t_i with the term as the attribute and the attribute value as its TFIDF weight [34], which is a statistical way of determining the relevance of a word to a document in a corpus. The most commonly used method to weight a term is the (TF.IDF) weighting, because it considers the attribute. With this weighting scheme, setting the weight of the term I in the document d is proportional to the number of times the term appears in the document, the Term Frequency (TF), and inversely related to the total number of documents in which the term appeared from the corpus, the Inverse Document Frequency (IDF).

The TFIDF weighting method assigns a weight to the number of term occurrences in a document by disregarding its relevance in case it appeared in most of the documents, especially when the term is assumed to possess little discriminating power:

$$w\_i = t f\_i \cdot \log\left(\frac{\mathrm{N}}{n}\right). \tag{4}$$

7.1. *Construction of the Three Classifiers.* In this experiment, the Arabic dataset documents were categorized using the following classifiers: KNN, NB, and DT in two forms, the full word (without stemming) and the stem word (full word stemmed by light10 stemmer).

7.2. *Evaluation and Comparison of Classification Quality.* Two measures are mainly used to evaluate the quality of the output of a classifier, namely, the f-measurement and accuracy [35]. In classification problems, the evaluation is generally represented in the form of a confusion matrix. The matrix contains the number of instances that are correctly and wrongly classified for each class.

In practice, the most widely used evaluation metric is the accuracy (ACC) rate. It represents the classifier efficiency based on the proportion of the number of correctly predicted instances the classifier made. The classifier accuracy is calculated as

$$\mathrm{ACC} = \left(\frac{(\mathrm{TP} + \mathrm{TN})}{(\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN})}\right). \tag{5}$$

Table 2: Sample showing different words derived from the same root, *Ktb* كتب.

Word in Arabic	Meaning in English	Root (stem) in Arabic	Root (stem) in English
مكتب	Office	كتب	*Ktb*
كاتب	Writer	كتب	*Ktb*
مكتبة	Library	كتب	*Ktb*
مكتوب	Written	كتب	*Ktb*

Table 3: List of prefixes and suffixes eliminated by Light10.

Prefixes	Suffixes
وال، بال، كال، فال، لل، و ،ال	ها، ان، ات، ون، ين، يه، ية، ه، ة، ي

Table 4: Examples of affixes in Arabic word (ليخادعونهم ,يرونهم).

Postfix	Suffix	Root	Prefix	Antefix
هم	ون	يرى	ي	ل
هم	ون	خدع	ي	ل
Pronoun meaning (they)	Termination of conjugation	يرى meaning (see) and خدع meaning (deceive)	A letter meaning the tense and the person of conjugation	Preposition meaning (to)

Table 5: Performance of the three classifiers without stemmer.

Classifiers	Number of features without stemmer	Accuracy (%)
DT	91756	90.2985
KNN	91756	26.119403
NB	91756	33.830846

Table 6: Performance of the three classifiers with stemmer.

Classifiers	Number of features before stemmer	Number of features after stemmer	Accuracy (%)
DT	91756	46167	93.7811
KNN	91756	46167	26.368159
NB	91756	46167	35.074627

8. Results

A comparison of the three classifiers was conducted in respect of accuracy and the number of features selected with and without the use of stemming in the preprocessing phase. Tables 5 and 6 show the results for the three classifiers with and without stemmer, respectively.

The tables show that, without a stemmer, DT outperformed KNN and NB achieving 90% accuracy as compared to 33.83% and 26.11%, respectively. When a stemmer was included in the preprocessing phase, all three classifiers improved their performance, and again, DT produced the best result with 93% as compared to NB with 35% and KNN with 26.36%. Thus, the use of a stemmer improved the accuracy of all three classifiers. Furthermore, the tables show that the use of a stemmer also reduced the number of features around 50% by the classifiers. Figure 5 provides a graphical illustration of the results, by which we can conclude that the number of features has effect on the NB and KNN performance. KNN when using all features got accuracy of 26.12, while when using stemmer the performance was not satisfying, with accuracy of 26.36%. On the other classifier of NB the stemmer enhances around 1.8% but comparing with DT the performance was better. We can conclude that the DT can be used for huge features better than NB and KNN.

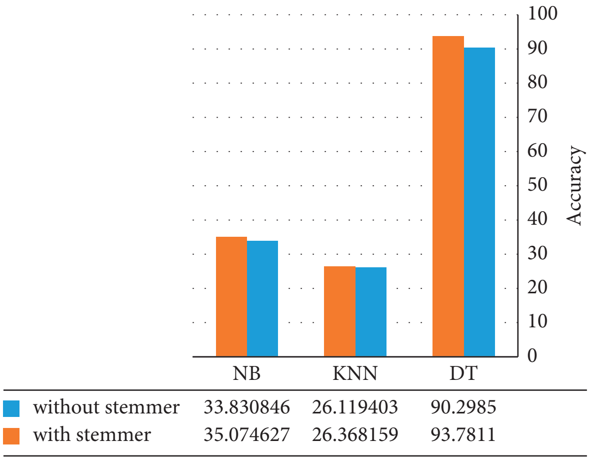


Figure 5: Accuracy of the three classifiers with/without stemmer.

The result shows that the decision tree with light stemmer was the best accuracy rate for classification algorithm with 93%.

9. Conclusion and Future Work

In this paper, prior to developing our proposed method, we reviewed several previous studies that contributed to improving our understanding of the study problem, namely, the classification of Arabic text, and potential solutions. Given the vast amount of information in Arabic that is available online, and which continues to grow, the main aim of this study was to save the effort and cost of both users and developers in searching for and using such data. In this work, we address the weakness of classifiers used for TC before as KNN, NB, and DT. The main weakness of the classifier algorithms is being poor when holding a huge number of features. Based on our experimental outcomes, we find that DT with stemmer can improve efficiency and outperform other classifiers compared to this work. However, the dimensionality of the terms without light stemming is the primary weakness in preprocessing phase, where there is a need for feature selection to fill the gap in the number of huge terms as a future work. We offer future work to improve text classifier with deep reinforcement Q-learning combined with our proposals. We also recommend the use of other classification criteria not used here in this work.

Data Availability

The data are available at https://catalog.ldc.upenn.edu/LDC2001T55 and are not free to access.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this study.

References

[1] J. McCarthy, M. L. Minsky, N. Rochester, and C. E. Shannon, "A proposal for the dartmouth summer research project on artificial intelligence, august 31, 1955," *AI Magazine*, vol. 27, no. 4, p. 12, 2006.

[2] M. M. Mafarja and S. Mirjalili, "Hybrid whale optimization algorithm with simulated annealing for feature selection," *Neurocomputing*, vol. 260, pp. 302–312, 2017.

[3] J. Han, J. Pei, and M. Kamber, *Data Mining: Concepts and Techniques*, Elsevier, Amsterdam, Netherlands, 2011.

[4] H. Drucker, D. Donghui Wu, and V. N. Vapnik, "Support vector machines for spam categorization," *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 1048–1054, 1999.

[5] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?" in *Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, vol. 10, pp. 79–86, July 2002.

[6] S. Saha, S. DasGupta, and S. K. Das, *Spam Mail Detection Using Data Mining: A Comparative Analysis, Smart Intelligent Computing and Applications, Smart Intelligent Computing and Applications*, S. Satapathy, V. Bhateja, and S. Das, Eds., vol. 104, pp. 571–580, Springer, Singapore, 2019.

[7] S. K. Trivedi and P. K. Panigrahi, "Spam classification: a comparative analysis of different boosted decision tree approaches," *Journal of Systems and Information Technology*, vol. 20, no. 3, pp. 298–105, 2018.

[8] S. Wang and J. Jiang, "Machine comprehension using match-lstm and answer pointer," arXiv preprint arXiv:1608.07905, 2016.

[9] A. Gupta, B. Eysenbach, C. Finn, and S. Levine, "Unsupervised meta-learning for reinforcement learning," arXiv preprint arXiv:1806.04640, 2018.

[10] J. Atwan, M. Wedyan, Q. Bsoul, A. Hamadeen, R. Alturki, and M. Ikram, "The effect of using light stemming for Arabic text classification," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 5, 2021.

[11] M. Davy and S. Luz, "Dimensionality reduction for active learning with nearest neighbour classifier in text categorisation problems," in *Proceedings of the Sixth International Conference on Machine Learning and Applications (ICMLA 2007)*, pp. 292–297, IEEE, Cincinnati, OH, USA, Dec. 2007.

[12] D. S. Guru, M. Ali, and M. Suhil, "A novel feature selection technique for text classification," *Advances in Intelligent Systems and Computing*, Springer, in *Proceedings of the Emerging Technologies in Data Mining and Information Security*, pp. 721–733, September 2018.

[13] 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*, pp. 51–58, New yark, NY,USA, August 2004.

[14] S. Eyheramendy, D. D. Lewis, and D. Madigan, "On the Naive Bayes Model for Text Categorization," in *Proceedings of the International Workshop on Artificial Intelligence and Statistics*, pp. 93–100, Key West, FL, USA, 2003.

[15] S. Al-Harbi, A. Almuhareb, A. Al-Thubaity, M. Khorsheed, and A. Al-Rajeh, "Automatic Arabic Text Classification," in *Proceedings of the 9th International Journal of Statistical Analysis of Textual Data*, pp. 77–83, Lyon, France, 2008.

[16] S. Jain, S. Shukla, and R. Wadhvani, "Dynamic selection of normalization techniques using data complexity measures," *Expert Systems with Applications*, vol. 106, pp. 252–262, 2018.

[17] S. Bahassine, A. Madani, and M. Kissi, "Arabic text classification using new stemmer for feature selection and decision trees," *Journal of Engineering Science & Technology*, vol. 12, no. 6, pp. 1475–1487, 2017.

[18] 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.

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

[20] H. T. Ng, W. B. Goh, and K. L. Low, "Feature selection, perceptron learning, and a usability case study for text categorization, ACM SIGIR Forum," in *Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval*, vol. 31, pp. 67–73, New yark, NY,USA, July 1997.

[21] W. e. Hadi, Q. A. Al-Radaideh, and S. Alhawari, "Integrating associative rule-based classification with Naïve Bayes for text classification," *Applied Soft Computing*, vol. 69, pp. 344–356, 2018.

[22] F. Colas and P. Brazdil, "Comparison of SVM and some older classification algorithms in text classification tasks,"vol. 207, pp. 169–178, in *Proceedings of the IFIP International*

Conference on Artificial Intelligence in Theory and Practice, vol. 207, Springer, Boston, MA, USA, August 2016.

[23] S. Bahassine, A. Madani, and M. Kissi, "Comparative study of Arabic text categorization using feature selection techniques and four classifier models," in *Proceedings of the 13th International Conference on Intelligent Systems: Theories and Applications*, pp. 1–5, ACMDL, New yark, NY,USA, September 2020.

[24] Y. Yang and X. Liu, "A re-examination of text categorization methods," in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 42–49, ACMDL, New yark, NY,USA, August 1999.

[25] Y. H. Li and A. K. Jain, "Classification of text documents," *The Computer Journal*, vol. 41, no. 8, pp. 537–546, 1998.

[26] S. A. Salloum, A. Q. AlHamad, M. Al-Emran, and K. Shaalan, "A survey of Arabic text mining," in *Proceedings of the Intelligent Natural Language Processing: Trends and Applications*, pp. 417–431, Springer, Heidelber, January 2018.

[27] 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.

[28] J. Atwan, M. Mohd, and G. Kanaan, "Enhanced Arabic information retrieval: light stemming and stop words," in *Proceedings of the International Multi-Conference on Artificial Intelligence Technology*, pp. 219–228, Springer, Berlin, Heidelberg, August 2013.

[29] A. Cole, D. Graff, and K. Walker, *Arabic Newswire Part 1 Corpus*, Linguistic Data Consortium (LDC), Philadelphia, PA, USA, 2001.

[30] F. C. Gey and D. W. Oard, "The TREC-2001 cross-language information retrieval track: searching Arabic using English, French or Arabic queries," *TREC*, vol. 2001, 26 pages, 2001.

[31] W. B. Croft, D. Metzler, and T. Strohman, "Search engines: information retrieval in practice," *Addison-Wesley Reading*, 2010.

[32] Q. Bsoul, E. Al-Shamari, M. Mohd, and J. Atwan, "Distance measures and stemming impact on Arabic document clustering," in *Proceedings of the Asia Information Retrieval Symposium*, pp. 327–339, Springer, Beiging, China, 18 November 2017.

[33] J. Atwan, M. Mohd, G. Kanaan, and Q. Bsoul, "Impact of stemmer on Arabic text retrieval, information retrieval technology," in *Proceedings of the Asia Information Retrieval Symposium*, vol. 8870, pp. 314–326, Springer, Cham.

[34] 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.

[35] M. S. G. Karypis, V. Kumar, and M. Steinbach, "A comparison of document clustering techniques," in *Proceedings of the TextMining Workshop at KDD2000*, Boston, MA, USA, May 2000.