An improved Arabic text classification method using word embedding

Tarik Sabri[1], Said Bahassine[2], Omar El Beggar[1], Mohamed Kissi[1]

[1]LIM Laboratory, Informatics Department, Faculty of Sciences and Techniques of Mohammedia, Hassan II University, Casablanca, Morocco

[2]Laboratory of Artificial Intelligence and Complex Systems Engineering, Department of Computer Science, National Higher School of Arts and Crafts, Hassan II University, Casablanca, Morocco

Article Info

Article history:

Received Apr 29, 2023
Revised Aug 8, 2023
Accepted Sep 6, 2023

Keywords:

Arabic language
Document representation
Feature selection
Text classification
Word embedding
Word2Vec

ABSTRACT

Feature selection (FS) is a widely used method for removing redundant or irrelevant features to improve classification accuracy and decrease the model's computational cost. In this paper, we present an improved method (referred to hereafter as RARF) for Arabic text classification (ATC) that employs the term frequency-inverse document frequency (TF-IDF) and Word2Vec embedding technique to identify words that have a particular semantic relationship. In addition, we have compared our method with four benchmark FS methods namely principal component analysis (PCA), linear discriminant analysis (LDA), chi-square, and mutual information (MI). Support vector machine (SVM), k-nearest neighbors (K-NN), and naive Bayes (NB) are three machine learning based algorithms used in this work. Two different Arabic datasets are utilized to perform a comparative analysis of these algorithms. This paper also evaluates the efficiency of our method for ATC on the basis of performance metrics viz accuracy, precision, recall, and F-measure. Results revealed that the highest accuracy achieved for the SVM classifier applied to the Khaleej-2004 Arabic dataset with 94.75%, while the same classifier recorded an accuracy of 94.01% for the Watan-2004 Arabic dataset.

Corresponding Author:

Tarik Sabri
LIM Laboratory, Informatics Department, Faculty of Sciences and Techniques of Mohammedia, Hassan II University
B.P. 146 Yasmina str. 20658 Mohammedia, Casablanca, Morocco
Email: sabritarik@gmail.com

1. INTRODUCTION

Text classification is one of the most common fields in text mining that associates a given text with one or more categories from a predefined set [1]. Each document is represented by a huge number of features that define the dimensionality of the dataset, making the process of training classification models difficult and slow [2]. This problem is known as the feature selection, which is the process of improving model performance by eliminating irrelevant and redundant features. Irrelevant features contain no interesting information on the topic of classification, whereas redundant features contain information that already exists in more useful features [3]. Common feature selection approaches in text classification include sentiment analysis [4]–[6], text classification [7], [8], image retrieval [9], and more. It is possible to select the most compelling features from the original datasets using a variety of feature selection approaches. In fact, there are three feature selection strategies: filter, wrapper, and hybrid based [10]. Each of these strategies decreases the number of features used while improving the results' precision. Although much research on feature selection (FS) for text classification has been conducted for languages such as English, German, Spanish, and Turkish [11], the number of publications dealing with Arabic remains limited due to it is morphology and grammatical rules [12].

Most of the FS methods have been proposed for text classification research. This can be done through i) filter-based (also known as traditional) FS techniques, such as mutual information (MI) [13], information gain [14], and Chi-square [7], [15]; ii) wrapper-based FS uses the ranking of the available features in terms of their relative importance to select a set of features to be used in the model. It can be time-consuming and computationally expensive, but it can produce powerful models that efficiently use the available data. This technique includes recursive feature elimination, sequential backward selection, and genetic algorithms [16], [17]; and iii) hybrid FS combines both filter and wrapper methods, the idea is to use a filter method to pre-select the most promising features before applying a wrapper method to further refine the selection [18], [19]. In recent years, many methods have been developed to extract the most relevant and meaningful features. We will look at some of the key methods in the following.

Tubishat *et al.* [4] proposed an improved feature selection method applied to Arabic sentiment analysis; namely, improved whale optimization algorithm (IWOA). They mixed information gain (IG) with whale optimization algorithm (WOA) using the support vector machine (SVM) classifier. Four datasets were used in the experiment: opinion corpus for Arabic (OCA), Arabic Twitter, political, and software. The findings revealed that the proposed method is more successful than five machine learning algorithms and two deep learning techniques such as: differential evolution (DE), grasshopper optimization algorithm (GOA), whale optimization algorithm (WOA), particle swarm optimization (PSO), genetic algorithm (GA), long short-term memory (LSTM), and convolutional neural network (CNN). The best accuracy obtained for this method compared to machine learning algorithms is 95.93%, while the best accuracy obtained compared to deep learning methods is 99.39%.

Marie-Sainte and Alalyani [20] have suggested a novel approach to improve the Arabic text classification (ATC) procedure called firefly algorithm based feature selection (FAFS), which is inspired by the firefly social behavior. Three evaluation metrics, including precision, recall and F-measure, are utilized in conjunction with the SVM classifier. The experimental tests showed that the FAFS method achieved a precision of 99.40%. In a similar vein, Singh *et al.* [21] have suggested another new method to improve the text classification accuracy that combines a term frequency-inverse document frequency (TF-IDF) with a Glove word embedding to identify words with similar semantic content. The most representative term with similar meanings is chosen as the one with the highest sum of TF-IDF. The authors also presented a new metric to evaluate the performance of the classifier on the reduced features. The results revealed that the suggested approach was more effective than principal component analysis (PCA), linear discriminant analysis (LDA), latent semantic indexing (LSI) and PCA+LDA. The authors have used three different corpora, namely: British Broadcasting Corporation (BBC), Classic4, and 20 newsgroups. The proposed method performed an accuracy of 96.18% on the BBC dataset, 91.12% for the Classic4 corpus, and 90.25% for the 20 newsgroups dataset.

Jin *et al.* [22] have developed a system for computing semantic similarity between words based on Word2Vec. For this purpose, the authors combined the word vector model, HowNet and TongYiCi CiLin, to compare the similarity of words. To enhance the similarity process and extend the coverage of all features, they improved the dictionaries and increased the size of the corpus to train the Word2Vec model. Sabri *et al.* [23] compared three feature vectorization techniques: TF-IDF, word count and Word2Vec. They have employed five different classifiers support vector machine (SVM), k-nearest neighbors (KNN), decision tree (DT), random forest (RF) and logistic regression. The experiments were applied to two common Arabic corpora: Arabic-CNN and OSAC-utf8. The study revealed that SVM and logistic regression models were more successful than all the other machine learning methods. The testing phase indicated that the vectorization method had a significant impact on increasing classification accuracy.

The effect of three algorithms, namely: naive Bayes (NB), k-nearest neighbors (KNN), and support vector machine (SVM) on spam email were studied comparatively by [24]. They improved the spam classification quality by reducing the number of features for classifiers using four optimization feature selection methods: genetic algorithm, harmony search, PSO, and local search. For experimentation, they used SPAM E-mail dataset with 4,601 emails and 1,813 spams. The models' performances were evaluated using both their accuracy and F-measure scores. The empirical results showed that SVM was more successful than other methods for spam classification when feature selection was integrated, whereas the NB classifier reported poorer results.

This paper proposes an improved FS method named removal of Arabic redundant features (RARF) to build feature subset from original Arabic dataset as well as improve model accuracy. First, we have generated our Word2Vec embedding model from five Arabic datasets with different sizes and classes. This model is able to compute the similarity between Arabic words. The second step involves grouping similar

Arabic words by multiplying it Is TF-IDF scores by the similarity values produced by our Word2Vec model. Experimental results on two publicly available Arabic datasets, namely Khaleej-2004 [25], and Watan-2004 [26], show that the proposed method gives better classification results than benchmark FS techniques, like PCA, LDA, Chi-square, and MI.

The remaining part of the paper is structured as follows: details of the proposed RARF method have been explained in the second 2. Section 3 is devoted to the experimental results and analysis. The last section includes a conclusion and recommendations for future work

2. RESEARCH METHOD

In this paper, we propose an improved FS technique that utilizes the word embedding method Word2Vec for ATC. Our contribution consists of identifying and grouping similar Arabic words based on the numerical vectors generated by our Word2Vec model. Arabic words in the same group are considered redundant features and each word group is replaced by a representative term obtained by TF-IDF weighted Word2Vec embedding. This helps to eliminate Arabic redundant features and can increase the classification accuracy. When using the dimension reduction techniques such as PCA, LDA, Chi-square, and MI, it is imperative to specify the final features of new vector space. In the case of our method, however, the number of dimensions depends on the groups obtained when applying the RARF algorithm.

The process of the proposed method involves several steps, as illustrated in Figure 1. This research explores three main stages: FS using our Word2Vec model created from five Arabic datasets, Arabic text classification, and prediction of the category of new Arabic document accurately. The following sub-sections discuss the design steps of the suggested method.

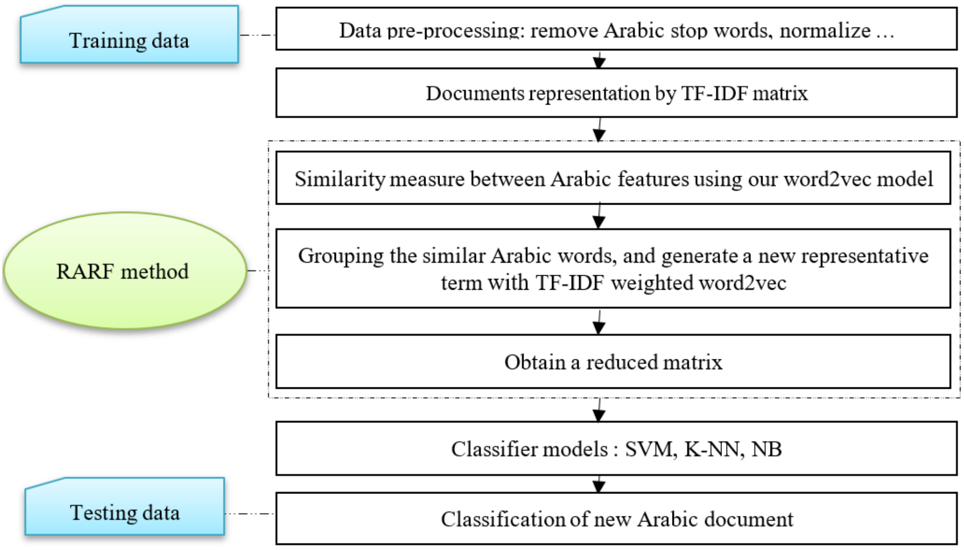
Training data
Data pre-processing: remove Arabic stop words, normalize …
Documents representation by TF-IDF matrix
Similarity measure between Arabic features using our word2vec model
RARF method
Grouping the similar Arabic words, and generate a new representative term with TF-IDF weighted word2vec
Obtain a reduced matrix
Classifier models : SVM, K-NN, NB
Testing data
Classification of new Arabic document

Figure 1. The main structure of the proposed method

2.1. Data pre-processing

Prior to classification, it is essential to undertake pre-processing of text data. In fact, the models used to classify texts will not be able to accurately identify the text's content if it is not given in an expected format. Each Arabic document in the dataset was processed in the following steps:

- Retire URLs, non-Arabic characters and symbols.
- Remove diacritics (for example, change the letter "بَ" to "ب").
- Remove extra whitespace and punctuation marks.
- Delete stopwords like "كيف", "من", "حتى" and non-meaningful words.
- Normalize "إأآٱ" to "ا" and "ي ئ" to "ى".

2.2. Documents representation by TF-IDF

In a vector space model, each document is considered as a set of numerical values in the vector space. The number of unique features means the number of dimensions. In this work, we use the TF-IDF as a bag of words (BoW) strategy to manage each document in the dataset. We have set a maximum number of features at 3,000, which allows us to select only the relevant features and to minimize the feature space. But this is not enough, we also need to eliminate the similar features. For this purpose, we use the Word2Vec as a technique to extract similar words.

2.3. Similarity detection using Word2Vec model

Word embedding is used to represent words or sentences in a text as vectors of numerical values [27]. These novel ways of representing text data have enabled an advancement in the accuracy of natural language processing (NLP) techniques, such as text classification. Word embedding is based on the linguistic concept of distributional semantics, which was pioneered by Harris [28]. This theory suggests that the meaning of a word is determined by its context. Therefore, words used in close contexts tend to have similar meanings. Word2Vec is one of the most widely known word embedding algorithms. The research project was led by Tomas Mikolov and conducted by a Google research team [29]. It is a double layer neural network-based algorithm that tries to learn the vector representations of words in a text. Word2Vec has two neural architectures, called continuous bag of words (CBOW) and Skip-Gram, from which the user can choose. Figure 2 shows the difference between the two models. CBOW takes the words in the context of a sentence and attempts to identify the target word, while skip-gram does the opposite and tries to predict the words that are in the context of the given word.

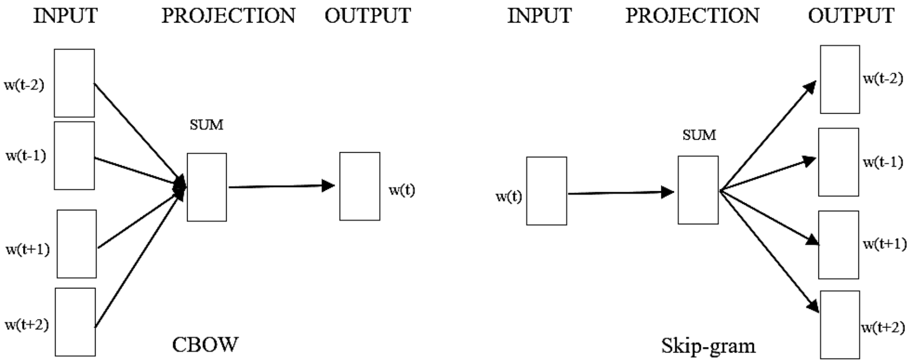

Figure 2. Word2Vec architectures [29]

To measure the similarity between features, we trained five Arabic datasets of different sizes and classes in order to create our own Word2Vec model. For this purpose, CBOW architecture was used because it is generally faster to train than skip-gram architecture and has better accuracy for frequent features [30]. This model generated 147,873 feature vectors with 300 dimensions, which would be used in the Arabic feature similarity measure. After this step, we chose a similarity threshold of 0.7 (between 0 and 1), because our empirical findings indicated that the classification can achieve higher accuracy results when using this threshold. If the similarity measure between the Arabic words exceeded this threshold, the words were grouped into the same set. Finally, we represented each set by TF-IDF scores weighted Word2Vec.

To train the Word2Vec model, a large series of experiments have been performed to adjust the hyperparameters (window size, vector size, and minimum count). We trained a batch of documents from five Arabic datasets using generate similar (Gensim) tools developed by Rehurek and Sojka [31]. It is a free and robust module for NLP which is used to generate word and document vectors. Table 1 shows the hyperparameters used to train our model as well as the Arabic datasets used.

Table 1. Word2Vec hyperparameters and datasets used to train the model

Architecture	Windows size	Vector size	Minimum count	Documents	Features	Arabic datasets

CBOW	5	300	3	59,903	147,873	Khaleej-2004 [25]; Watan-2004 [26];
						ANTCorpus [32];
						Arabic-CNN [33]; OSAC [33]

2.4. Dimension reduction methods

Dimension reduction in text classification is a technique used to reduce complexity by minimizing the number of features used to train a text classification model. This can be accomplished by selecting the most relevant features and removing redundancies. To evaluate the performance of our method, the following four well-known methods were used: PCA, LDA, Chi-square, and MI. PCA is a statistical method used in text classification that reduces the number of features by transforming a set of related variables into a smaller set of variables that explain most of the variance in the original set [34]. LDA uses a probabilistic graphical model to infer topics from a dataset by analyzing word frequency distributions [35]. The topics inferred by LDA can then be used as features for supervised learning algorithms in order to predict the class of a given text. The Chi-square test is used in statistics to test the independence of two events, it can select features that are the most likely to have an impact on the target variable and allow the model to be more accurate [7]. MI is a selection method that measures the dependence of an independent variable on the target variable. As such, MI is zero when two variables are independent, while a higher value reflects a greater dependence [36].

2.5. Removal of Arabic redundant features

In this section, we describe of our proposed method for Arabic text classification. It makes use the Word2Vec embedding technique and term weighting TF-IDF. The aim is to find features with certain semantic relationships. The method steps could be synthesized by algorithm 1.

Algorithm 1. RARF

```
Input:
$D=\{d_1,d_2,d_3,...,d_n\}$: the dataset; $F=\{f_1,f_2,f_3,...,f_m\}$: the features
$M=\{m_{ij}=TF\text{-}IDF(d_i,f_j)\}$ represents the matrix of TF-IDF features
model_w2v: the Word2Vec model
α: the predetermined similarity threshold (experimentally, we have chosen 0.7 as the value
of α).
Output:
F′: an optimal subset of features.
Steps:
1.  for each $f_i$ in F do
2.          group_words={}
3.          for each $f_{i+1}$ in F do
4.                  similarity=model_w2v(vect($f_i$),vect($f_{i+1}$))
5.                  if similarity > α do
6.                          group_words=group_words U {$f_i$, $f_{i+1}$}
7.                  end if
8.          end for
9.          generate a single representative feature $RF_i$ from $i^{th}$ group_words using TF-
    IDF scores weighted Word2vec
10.         F′=F′ U {$RF_i$}
11. end for
12. return feature set F′
13. F′={$RF_1$,$RF_2$,$RF_3$…...,$RF_k$}, where k<m and $RF_k$ is the representative feature of $k^{th}$
    group_words
```

Example:

Table 2 shows the TF-IDF matrix M in a dataset contains n documents using m features. m_{ij} represents a statistical metric that evaluates the improtance of feature i in document d_i relative to the dataset. We suppose that the i^{th} group contains the features: f_1, f_4, f_6 and f_7: $group\_words_i = \{f_1, f_4, f_6, f_7\}$. Table 3 represents the similarities between words of the i^{th} group. Where $sim\_value\_f_1\_f_4$ is the similarity value between first and 4th feature.

Table 2. TF-IDF representation

$M=\{m_{ij}=TF\text{-}IDF(d_i,f_j)\}$	f_1	f_2	f_3	f_4	f_5	f_6	f_7	…	f_m
d_1	m_{11}	m_{12}	m_{13}	m_{14}	m_{15}	m_{16}	m_{17}	…	m_{1m}
d_2	m_{21}	m_{22}	m_{23}	m_{24}	m_{25}	m_{26}	m_{27}	…	m_{2m}
…	…	…	…	…	…	…	…	…	…
d_n	m_{n1}	m_{n2}	m_{n3}	m_{n4}	m_{n5}	m_{n6}	m_{n7}	…	m_{nm}

Table 3. Example of similarity rates between words of the same group

similarity	f_4	f_6	f_7
f_1	$sim\_value\_f_1\_f_4$	$sim\_value\_f_1\_f_6$	$sim\_value\_f_1\_f_7$

2.5.1. The representative feature RF_i for $group\_words_i$

To calculate the vector values that represents the i^{th} representative feature (RF_i), we used both the values produced by the TF-IDF matrix and the Word2Vec similarities. It is represented by (1):

$$RF_i = \begin{pmatrix} RF_{i1} \\ RF_{i2} \\ RF_{i3} \\ \dots \\ \dots \\ RF_{in} \end{pmatrix} = \begin{pmatrix} m_{11} \\ m_{21} \\ m_{31} \\ \dots \\ \dots \\ m_{n1} \end{pmatrix} \times sim\_value\_f_1\_f_1 + \begin{pmatrix} m_{14} \\ m_{24} \\ m_{34} \\ \dots \\ \dots \\ m_{n4} \end{pmatrix} \times sim\_value\_f_1\_f_4 + \begin{pmatrix} m_{16} \\ m_{26} \\ m_{36} \\ \dots \\ \dots \\ m_{n6} \end{pmatrix} \times sim\_value\_f_1\_f_6 + \begin{pmatrix} m_{17} \\ m_{27} \\ m_{37} \\ \dots \\ \dots \\ m_{n7} \end{pmatrix} \times sim\_value\_f_1\_f_7 \quad (1)$$

$$RF_i = \begin{pmatrix} RF_{i1} \\ RF_{i2} \\ RF_{i3} \\ \dots \\ \dots \\ RF_{in} \end{pmatrix} = \begin{pmatrix} m_{11} \\ m_{21} \\ m_{31} \\ \dots \\ \dots \\ m_{n1} \end{pmatrix} + \begin{pmatrix} m_{14} \\ m_{24} \\ m_{34} \\ \dots \\ \dots \\ m_{n4} \end{pmatrix} \times sim\_value\_f_1\_f_4 + \begin{pmatrix} m_{16} \\ m_{26} \\ m_{36} \\ \dots \\ \dots \\ m_{n6} \end{pmatrix} \times sim\_value\_f_1\_f_6 + \begin{pmatrix} m_{17} \\ m_{27} \\ m_{37} \\ \dots \\ \dots \\ m_{n7} \end{pmatrix} \times sim\_value\_f_1\_f_7 \;;\; where\colon sim\_value\_f_1\_f_1 = 1$$

$$RF_{i1} = m_{11} + (m_{14} \times sim\_value\_f_1\_f_4) + (m_{16} \times sim\_value\_f_1\_f_6) + (m_{17} \times sim\_value\_f_1\_f_7)$$

$$RF_{i2} = m_{21} + (m_{24} \times sim\_value\_f_1\_f_4) + (m_{26} \times sim\_value\_f_1\_f_6) + (m_{27} \times sim\_value\_f_1\_f_7)$$

$$RF_{in} = m_{n1} + (m_{n4} \times sim\_value\_f_1\_f_4) + (m_{n6} \times sim\_value\_f_1\_f_6) + (m_{n7} \times sim\_value\_f_1\_f_7)$$

$$RF_i = [RF_{i1}, RF_{i2}, RF_{i3}, \dots, RF_{in}];\; F' = \{RF_1, RF_2, \dots, RF_k\}; k < m$$

Finally, we replace all the terms of this group (for this example: f_1, f_4, f_6 and f_7) by RF_i in the matrix.

2.6. How does the proposed method differ from existing methods?

The RARF FS method groups similar words by combining TF-IDF values and similarity rates generated by the Word2Vec model. To do this, it requires a simpler mathematical calculation compared to statistical-based methods. When applying the PCA, LDA, Chi-square, and MI FS methods, it is important to specify the number of features to be used, and therefore the features to be reduced. However, the number of features need not be specified for the RARF approach. The RARF method is based on a threshold (between 0 and 1); if the similarity measure between Arabic words exceeds an empirical value of 0.7, the words are grouped together in the same set. PCA can fail when the data is too complex, and it does not work with data that is highly imbalanced. When the data is highly unbalanced LDA will not be able to learn much useful information from it. In multi-class datasets, LDA may struggle to separate classes and accurately classify new data points. Also, if the data does not follow a normal distribution, the Chi-square test will not be effective. MI is only suitable for discrete variables, so if the variables are continuous, then other methods need to be employed. In such cases, RARF can work well because imbalanced data, discrete variables, multi-class, and distribution are not blocking parameters for it.

2.7. Classifiers

Text classification is well known in many languages, especially in English [37]. Despite the importance of the Arabic language, little research has been conducted on Arabic text classification using the concept of similarities between Arabic features. Machine learning classifiers are essential for text classification because they provide an automated and powerful way to identify patterns in text documents. In this work, we have used SVM, KNN, and NB common classifiers to assess the efficiency of our proposed method. SVM are a series of machine learning algorithms that solve problems like classification and regression [38]. They divide data into distinct categories using the simplest boundary possible, in order to maximize the distance between the separate groups of data and the boundary that separates them. KNN is a standard classification algorithm that relies exclusively on the choice of the classification metric. The idea is the following: from a labeled database, we can estimate the class of a new data by looking at the majority class of the *k* closest neighboring data (hence the name of the algorithm) [39]. The only parameter to set is *k*, the number of neighbors to consider. NB classifier is a type of simple probabilistic Bayesian classification based on Bayes' theorem with a strong independence (called naive) of assumptions [40]. It uses a naive Bayes classifier, belonging to the family of linear classifiers.

3. RESULTS AND DISCUSSION

3.1. Data collection

Data collection can be described as a gathering of text-based documents that can be divided into various categories. We used two Arabic benchmark datasets with various numbers and sizes of categories to conduct our studies. The Khaleej-2004 dataset is a commonly used reference for Arabic datasets. It is a collection of 5,690 documents distributed in 4 classes: economy (909 documents), international news (953 documents), local news (2,398 documents) and sport (1,430 documents). The Watan-2004 is a large Arabic corpus containing 20,291 documents. Each document is tagged with one of the following six categories: culture (2,782 documents), economy (3,468 documents), international (2,035 documents), local (3,596 documents), religion (3,860 documents), and sports (4,550 documents). The datasets are divided into two parts: The training set contains 80% of the dataset documents while the test set represents 20%. The distribution of the training and test sets is represented in Table 4.

Table 4. Training set and test set for benchmark Arabic datasets

Arabic dataset	Training set	Test set	Total
Khaleej-2004	4552	1138	5690
Watan-2004	16232	4059	20291

3.2. Performance evaluation

The confusion matrix (also known as error matrix) is widely used to summarize the performance of a classifier model. It presents the numbers of real and predicted labels. This matrix is a two-dimensional table consisting of two columns and two rows that indicate four meaningful values: true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN) as shown in Table 5. The common metrics that can be measured from a confusion matrix are: accuracy, precision, recall and F-measure. These metrics are used to interpret the results of our method.

Table 5. Confusion matrix

	Predicted positive	Predicted negative
Actually positive	TP	FN
Actually negative	FP	TN

- Accuracy: it is the ratio of correct predictions to the total number of predictions. It can be defined by (2).

$$\text{accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \tag{2}$$

- Precision: it is the ratio of true positive to all predicted positive. It is also called positive predictive value (PPV). It can be defined by (3).

$$\text{precision} = \frac{\text{TP}}{\text{TP+FP}} \tag{3}$$

- Recall: it is the ratio of true positive to actual positive. It measures how sensitive a model is to the positive class. It can be defined by (4).

$$\text{recall} = \frac{\text{TP}}{\text{TP+FN}} \tag{4}$$

– F-measure: it is the harmonic mean of precision and recall and it is given by (5).

$$\text{F} - \text{measure} = \frac{2\times\text{precision}\times\text{recall}}{\text{precision}+\text{recall}} \tag{5}$$

3.3. Results and analysis

To make an effective comparison of our approach with the other FS methods, we have used three classifiers that work differently SVM, K-NN, and DT. The results were examined using frequently used evaluation measures: accuracy, precision, recall, and F-measure. For the Khaleej-2004 dataset, Tables 6, 7, and 8 represent the results of five FS methods PCA, LDA, Chi-square, MI, and RARF when SVM, K-NN, and DT machine learning models are used. The experiments were performed to assess the effectiveness of the RARF proposed approach compared with other FS methods for the three classifiers.

The calculated performance measures for the SVM classifier are shown in Table 6 which depicts that the RARF method outperformed all other feature selection techniques with a maximum accuracy of 94.75%, while Chi-square achieved the second highest accuracy of 94.20%. In addition, the highest precision of 94.07% is obtained by RARF, and the lowest precision of 92.96% is obtained by MI. The highest recall obtained by the techniques is 94.59%, and 94.26% in RARF, and Chi-square, respectively. RARF is outperformed by an F-measure of 94.30%. Chi-square, MI, and PCA techniques have an F-measure of nearly 93%, while the LDA scored the worst recall performance at 72.79%.

Table 7 shows the performance for Khaleej-2004 dataset using K-NN classifier. The accuracy obtained by the RARF is 92.51%, 92.36%, 92.01%, and 92.15% in accuracy, precision, recall, and F-measure, respectively, while MI achieved the second highest accuracy of 85.04% with a variation of 7.43% compared to the RARF. It should be noted that the LDA method achieved the lowest efficiency, close to 77% for all four metrics.

Table 8 summarizes the results achieved by the NB classifier. The first point to mention is that the RARF method has achieved the best performance with 88.10%, 88.67%, 87.72%, and 88.09% in accuracy, precision, recall, and F-measure, respectively. Though, the NB classifier underperforms compared with SVM and K-NN. The MI achieved the second highest performance of 83.30%, 87.54%, 82.53%, and 83.77% for accuracy, precision, recall, and F-measure, respectively. It should also be noted that the PCA offers the poorest performance, with a precision of 65.36%. To summarize, for the Khaleej-2004 dataset, the RARF performs better than PCA, LDA, Chi-square, and MI methods, the highest rate of accuracy, precision, recall, and F-measure is achieved when RARF is applied in conjunction with an SVM classifier, at nearly 94%, followed by the K-NN, at nearly 92%.

Table 6. Performance analysis for Khaleej-2004 dataset using SVM

Methods	Accuracy	Precision	Recall	F-measure
PCA	93.66	93.22	93.40	93.28
LDA	74.52	73.12	72.79	72.79
Chi-square	94.20	93.42	94.26	93.80
MI	93.94	92.96	94.21	93.52
RARF	**94.75**	**94.07**	**94.59**	**94.30**

Table 7. Performance analysis for Khaleej-2004 dataset using K-NN

Methods	Accuracy	Precision	Recall	F-measure
PCA	80.63	82.20	83.04	81.03
LDA	76.63	76.40	76.21	76.20
Chi-square	84.22	79.22	90.11	83.02
MI	85.04	80.61	89.80	83.92
RARF	**92.51**	**92.36**	**92.01**	**92.15**

Table 8. Performance analysis for Khaleej-2004 dataset using NB

Methods	Accuracy	Precision	Recall	F-measure
PCA	65.98	65.36	72.69	67.08
LDA	75.31	74.28	74.15	74.17
Chi-square	82.16	86.67	81.60	82.73
MI	83.30	87.54	82.53	83.77
RARF	**88.10**	**88.67**	**87.72**	**88.09**

Tables 9, 10, and 11 represent the results of the Watan-2004 dataset, Table 9 shows the accuracy, precision, recall, and F-measure of FS methods using SVM classifier. The accuracy obtained by the techniques is 92.15%, 67.90%, 92.89%, 92.68%, 94.01% in PCA, LDA, Chi-square, MI, and RARF, respectively. RARF achieves 93.60% better precision, while the LDA offers a minimum precision of 56.23%. The highest F-measure obtained by the techniques is 93.65%, and 92.50% in RARF, and Chi-square, respectively. Table 9 shows that RARF outperformed all the other FS techniques, with a highest recall of 93.97%, while Chi-square and MI achieved the second highest recall of 92.50% and 92.27%, respectively.

Table 10 shows the final scores of the evaluated FS methods. These results present the scores achieved by the base classifier K-NN. We can make several observations from these findings. First, the LDA obtained the lowest average performance score for the most experiments with a minimum precision of 54.79%. Second, all the RARF metrics have a score of nearly 87%, which is comparatively better, monitored by Chi-square with an accuracy of 85.73%.

Experimental results for the NB classifier are shown in Table 11. The RARF method has always proved its effectiveness in the classification process, with scores between 89% and 90%. Followed by the Chi-square with values close to 84%. We also found that the LDA classifier consistently delivers the worst results, with scores of less than 67%. For the Watan-2004 dataset, we can assume that the SVM classifier gives encouraging results compared to the K-NN, and NB classifiers. The RARF method generates significant results compared to the PCA, LDA, Chi-square and MI feature selection methods. Additionally, we can note that the LDA is producing unsatisfactory results.

Table 9. Performance analysis for Watan-2004 dataset using SVM

Methods	Accuracy	Precision	Recall	F-measure
PCA	92.15	91.79	91.63	91.69
LDA	67.90	56.23	60.92	58.38
Chi-square	92.89	92.43	92.61	92.50
MI	92.68	92.21	92.37	92.27
RARF	**94.01**	**93.60**	**93.75**	**93.65**

Table 10. Performance analysis for Watan-2004 dataset using K-NN

Methods	Accuracy	Precision	Recall	F-measure
PCA	68.91	67.93	70.34	68.30
LDA	66.35	54.79	59.29	56.68
Chi-square	85.73	85.05	85.16	84.66
MI	84.43	83.76	84.06	83.36
RARF	**87.78**	**87.26**	**86.96**	**87.03**

Table 11. Performance analysis for Watan-2004 dataset using NB

Methods	Accuracy	Precision	Recall	F-measure
PCA	83.84	82.00	86.73	82.46
LDA	64.18	57.89	59.12	58.01
Chi-square	84.85	83.13	84.99	83.61
MI	82.40	80.05	84.21	81.02
RARF	**90.48**	**89.46**	**90.29**	**89.75**

4. CONCLUSION AND FUTURE WORK

In this work, we propose an improved method for removal Arabic redundant features (RARF) based on word embedding. First, we have built a Word2Vec model using five Arabic datasets. The Word2Vec model is able to group Arabic words with similar semantic meaning. Second, to reduce the feature numbers in BoW, the RARF method represents each word group by a single representative term using the TF-IDF scores weighted Word2Vec model vectors. The experiment indicated that employing the suggested approach for text classification yielded more successful outcomes than utilizing PCA, LDA, Chi-square and MI. We also observed that the SVM method performed best, while the LDA method performed worst. In upcoming studies, we will attempt to apply this method to other datasets of different languages in order to improve the generalization of our algorithm. In addition, we will try to train a Word2Vec model with more Arabic datasets, which will allow us to capture more semantic relations between Arabic words. Furthermore, we plan to apply the RARF algorithm to specific dialects in the Arab world, such as Moroccan and Egyptian dialects.

REFERENCE

[1] F. Hemmatian and M. K. Sohrabi, “A survey on classification techniques for opinion mining and sentiment analysis,” *Artificial Intelligence Review*, vol. 52, no. 3, pp. 1495–1545, Oct. 2019, doi: 10.1007/s10462-017-9599-6.

[2] A. H. Hossny, L. Mitchell, N. Lothian, and G. Osborne, “Feature selection methods for event detection in Twitter: a text mining approach,” *Social Network Analysis and Mining*, vol. 10, no. 1, Dec. 2020, doi: 10.1007/s13278-020-00658-3.

[3] S. Chormunge and S. Jena, “Efficient feature subset selection algorithm for high dimensional data,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 6, no. 4, Aug. 2016, doi: 10.11591/ijece.v6i4.9800.

[4] M. Tubishat, M. A. M. Abushariah, N. Idris, and I. Aljarah, “Improved whale optimization algorithm for feature selection in Arabic sentiment analysis,” *Applied Intelligence*, vol. 49, no. 5, pp. 1688–1707, May 2019, doi: 10.1007/s10489-018-1334-8.

[5] F. Akbarian and F. Z. Boroujeni, “An improved feature selection method for sentiments analysis in social networks,” in *2020 10th International Conference on Computer and Knowledge Engineering (ICCKE)*, Oct. 2020, pp. 181–186, doi: 10.1109/ICCKE50421.2020.9303710.

[6] A. Shaddeli, F. S. Gharehchopogh, M. Masdari, and V. Solouk, “An improved African vulture optimization algorithm for feature selection problems and its application of sentiment analysis on movie reviews,” *Big Data and Cognitive Computing*, vol. 6, no. 4, Sep. 2022, doi: 10.3390/bdcc6040104.

[7] 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, Feb. 2020, doi: 10.1016/j.jksuci.2018.05.010.

[8] N. S. Mohd Nafis and S. Awang, “An enhanced hybrid feature selection technique using term frequency-inverse document frequency and support vector machine-recursive feature elimination for sentiment classification,” *IEEE Access*, vol. 9, pp. 52177–52192, 2021, doi: 10.1109/ACCESS.2021.3069001.

[9] S. Uyun and L. Choridah, “Feature selection mammogram based on breast cancer mining,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 8, no. 1, pp. 60–69, Feb. 2018, doi: 10.11591/ijece.v8i1.pp60-69.

[10] Y. Guo, F.-L. Chung, G. Li, and L. Zhang, “Multi-label bioinformatics data classification with ensemble embedded feature selection,” *IEEE Access*, vol. 7, pp. 103863–103875, 2019, doi: 10.1109/ACCESS.2019.2931035.

[11] T. Parlar, S. A. Özel, and F. Song, “QER: a new feature selection method for sentiment analysis,” *Human-centric Computing and Information Sciences*, vol. 8, no. 1, Dec. 2018, doi: 10.1186/s13673-018-0135-8.

[12] 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, doi: 10.1109/ACCESS.2019.2903331.

[13] E. Shamoi, A. Turdybay, P. Shamoi, I. Akhmetov, A. Jaxylykova, and A. Pak, “Sentiment analysis of vegan related tweets using mutual information for feature selection,” *PeerJ Computer Science*, vol. 8, Dec. 2022, doi: 10.7717/peerj-cs.1149.

[14] W. Kaur, V. Balakrishnan, and K.-S. Wong, “Improving multi-label text classification using weighted information gain and co-trained multinomial naïve Bayes classifier,” *Malaysian Journal of Computer Science*, vol. 35, no. 1, pp. 21–36, Jan. 2022, doi: 10.22452/mjcs.vol35no1.2.

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

[16] A.-D. Li, B. Xue, and M. Zhang, “Improved binary particle swarm optimization for feature selection with new initialization and search space reduction strategies,” *Applied Soft Computing*, vol. 106, Jul. 2021, doi: 10.1016/j.asoc.2021.107302.

[17] R. Guha *et al.*, “Deluge based genetic algorithm for feature selection,” *Evolutionary Intelligence*, vol. 14, no. 2, pp. 357–367, Jun. 2021, doi: 10.1007/s12065-019-00218-5.

[18] M. M. Mafarja and S. Mirjalili, “Hybrid binary ant lion optimizer with rough set and approximate entropy reducts for feature selection,” *Soft Computing*, vol. 23, no. 15, pp. 6249–6265, Aug. 2019, doi: 10.1007/s00500-018-3282-y.

[19] W. BinSaeedan and S. Alramlawi, “CS-BPSO: Hybrid feature selection based on chi-square and binary PSO algorithm for Arabic email authorship analysis,” *Knowledge-Based Systems*, vol. 227, Sep. 2021, doi: 10.1016/j.knosys.2021.107224.

[20] S. L. Marie-Sainte and N. Alalyani, “Firefly algorithm based feature selection for Arabic text classification,” *Journal of King Saud University-Computer and Information Sciences*, vol. 32, no. 3, pp. 320–328, Mar. 2020, doi: 10.1016/j.jksuci.2018.06.004.

[21] K. N. Singh, S. D. Devi, H. M. Devi, and A. K. Mahanta, “A novel approach for dimension reduction using word embedding: An enhanced text classification approach,” *International Journal of Information Management Data Insights*, vol. 2, no. 1, Apr. 2022, doi: 10.1016/j.jjimei.2022.100061.

[22] X. Jin, S. Zhang, and J. Liu, “Word semantic similarity calculation based on Word2Vec,” in *2018 International Conference on Control, Automation and Information Sciences (ICCAIS)*, Oct. 2018, pp. 12–16, doi: 10.1109/ICCAIS.2018.8570612.

[23] T. Sabri, O. El Beggar, and M. Kissi, “Comparative study of Arabic text classification using feature vectorization methods,” *Procedia Computer Science*, vol. 198, pp. 269–275, 2022, doi: 10.1016/j.procs.2021.12.239.

[24] G. Rawashdeh, R. Mamat, Z. B. A. Bakar, and N. H. A. Rahim, “Comparative between optimization feature selection by using classifiers algorithms on spam email,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 6, pp. 5479–5485, Dec. 2019, doi: 10.11591/ijece.v9i6.pp5479-5485.

[25] M. Abbas and K. Smaili, “Comparison of topic identification methods for Arabic language,” in *Proceedings of International Conference on Recent Advances in Natural Language Processing, RANLP*, 2005, pp. 14–17.

[26] M. Abbas, K. Smaili, and D. Berkani, “Evaluation of topic identification methods on Arabic corpora,” *Journal of Digital Information Management*, vol. 9, no. 5, pp. 185–192, 2011.

[27] F. Torregrossa, R. Allesiardo, V. Claveau, N. Kooli, and G. Gravier, “A survey on training and evaluation of word embeddings,” *International Journal of Data Science and Analytics*, vol. 11, no. 2, pp. 85–103, Mar. 2021, doi: 10.1007/s41060-021-00242-8.

[28] Z. S. Harris, “Distributional structure,” *Word*, vol. 10, no. 2–3, pp. 146–162, Aug. 1954, doi: 10.1080/00437956.1954.11659520.

[29] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” *arXiv preprint arXiv:1301.3781*, Jan. 2013.

[30] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” *Advances in neural information processing systems*, vol. 26, Oct. 2013.

[31] R. Rehurek and P. Sojka, “Gensim-python framework for vector space modelling,” *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, vol. 3, no. 2, 2011.

[32] A. Chouigui, O. Ben Khiroun, and B. Elayeb, “ANT corpus: An Arabic news text collection for textual classification,” in *2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA)*, Oct. 2017, pp. 135–142, doi: 10.1109/AICCSA.2017.22.

[33] M. K. Saad and W. Ashour, “Osac: Open source Arabic corpora,” *International Conference on Electrical and Computer Systems (EECS’10)*, pp. 25–26, Nov. 2012.

[34] Y. Zhang, G. Li, and H. Zong, “A method of dimensionality reduction by selection of components in principal component analysis for text classification,” *Filomat*, vol. 32, no. 5, pp. 1499–1506, 2018, doi: 10.2298/FIL1805499Z.

[35] A. Panichella, “A systematic comparison of search-based approaches for LDA hyperparameter tuning,” *Information and Software Technology*, vol. 130, Feb. 2021, doi: 10.1016/j.infsof.2020.106411.

[36] X. Su and F. Liu, “A survey for study of feature selection based on mutual information,” in *2018 9th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, Sep. 2018, pp. 1–4, doi: 10.1109/WHISPERS.2018.8746913.

[37] A. Elnagar, R. Al-Debsi, and O. Einea, “Arabic text classification using deep learning models,” *Information Processing and Management*, vol. 57, no. 1, Jan. 2020, doi: 10.1016/j.ipm.2019.102121.

[38] E. Alickovic and A. Subasi, “Ensemble SVM method for automatic sleep stage classification,” *IEEE Transactions on Instrumentation and Measurement*, vol. 67, no. 6, pp. 1258–1265, Jun. 2018, doi: 10.1109/TIM.2018.2799059.

[39] J. Maillo, S. Ramírez, I. Triguero, and F. Herrera, “kNN-IS: an iterative spark-based design of the k-nearest neighbors classifier for big data,” *Knowledge-Based Systems*, vol. 117, pp. 3–15, Feb. 2017, doi: 10.1016/j.knosys.2016.06.012.

[40] Q. He *et al.*, “Landslide spatial modelling using novel bivariate statistical based Naïve bayes, RBF Classifier, and RBF network machine learning algorithms,” *Science of The Total Environment*, vol. 663, pp. 1–15, May 2019, doi: 10.1016/j.scitotenv.2019.01.329.

BIOGRAPHIES OF AUTHORS

Tarik Sabri is a Ph.D. student of computer science at Faculty of Sciences and Technology, Hassan II University, Morocco. He obtained his Master in data science and big data from ENSIAS, Mohamed V University, Rabat, Morocco in 2019. His research interests are natural language processing, machine learning, text mining (Arabic) especially text classification and feature selection. He can be contacted at email: sabritarik@gmail.com.

Said Bahassine received his Ph.D. degree from Faculty of Sciences, Chouaib Doukkali University, El Jadida, Morocco in 2019. He is currently a professor in Department of Computer Science, National Higher School of Arts and Crafts, Hassan II University, Casablanca, Morocco. Member of the Laboratory of Artificial Intelligence and Complex Systems Engineering (AICSE), his research interests include natural language processing, feature selection, machine learning and text mining. He is the author of many research papers published at conference proceedings and international journals. He can be contacted at email: said.bahassine@univh2c.ma.

Omar El Beggar received the Eng. degree in software engineering from ENSIAS, University Mohammed V, in 2002 and the Ph.D. degree in computer science from FSTS, University Hassan I, Morocco, in 2013. He obtained afterwards his HDR in soft computing and meta-modelling of decisional support systems at FSTM, University Hassan II, Morocco, in 2019. Currently, he is a full professor at the Department of Computer Science, FSTM, University Hassan II. Since 2021, he was the Pedagogical Director of the engineering department “Software Engineering and IT Systems Integration” (ILISI) within the same faculty. His research interests include green IT, explainable artificial intelligence, machine learning, MDA, multi-criteria decision aid, less/no code platforms. He has published several works in many indexed journals and international conferences. He can be contacted at email: omar.elbeggar@fstm.ac.ma.

Mohamed Kissi received his Ph.D. degree from the UPMC, France in 2004 in computer science. He is currently a full professor in Department of Computer Science, University Hassan II Casablanca, Faculty of Sciences and Technologies, Mohammedia, Morocco. His current research interests include machine learning, data and text mining (Arabic) and big data. He is the author of many research papers published at conference proceedings and international journals in Arabic text mining, bioinformatics, genetic algorithms and fuzzy sets and systems. He can be contacted at email: mohamed.kissi@fstm.ac.ma.