Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 05, No. 03, December 2019.

A PROPOSED MODEL OF SELECTING FEATURES FOR CLASSIFYING ARABIC TEXT

Ahmed M. D. E. Hassanein[1] and Mohamed Nour[2]

(Received: 25-Jul.-2019, Revised: 15-Oct.-2019, Accepted: 2-Nov.-2019)

ABSTRACT

Classification of Arabic text plays an important role for several applications. Text classification aims at assigning predefined classes to text documents. Unstructured Arabic text can be easily processed by humans, while it is harder to be interpreted and understood by machines. So, before classifying Arabic text or documents, some pre-processing operations should be done.

This work presents a proposed model for selecting features from the adopted Arabic text; i.e., documents. In this work, the words 'text' and 'documents' are used interchangeably. The adopted documents are taken from Al-Khaleej-2004 corpus. The corpus contains thousands of documents which talk about news in different domains, such as economics, as well as international, local and sport news. Some preprocessing operations are carried out to extract the highly weighted terms that best describe the content of the documents. The proposed model contains many steps to define the most relevant features. After defining the initial number of features, based on the weighted words, the steps of the model begin. The first step is based on calculating the correlation between each feature and class one. Depending on a threshold value, the most highly correlated features are chosen. This reduces the number of chosen features. The number of features is again reduced by calculating the intra-correlation between the resultant features. This is done in the second step. The third step selects the best features from among those which resulted from the second step by adopting some logical operations. The logical operations, specifically logical AND or logical OR, are applied to fuse the values of features depending on their structure, nature and semantics. The obtained features are then reduced in number. The fourth step is based on adopting the idea of document clustering; i.e., the obtained features from step three are placed in one cluster. Then, iterative operations are used to group features into two clusters. Each cluster can be further partitioned into two clusters ...and so on. That partitioning is repeated till the clusters' contents are not changed. The contents of each cluster are fused together using the cosine rule. This reduces the overall number of features.

This work adopts four types of classifiers; namely, Naïve Bayes (NB), Decision Tree, CART and KNN. A comparative study is carried out among the behaviors of the adopted classifiers on the selected number of features. The comparative study considers some measurable criteria; namely, precision, recall, F-measure and accuracy. This work is implemented using WEKA and MatLab software packages. From the obtained results, the best performance is achieved by using CART classifier, while the worst one is obtained by using KNN classifier.

KEYWORDS

Text classification, Text clustering, Feature selection, Arabic datasets, Machine learning methods, Performance evaluation.

1. INTRODUCTION AND RELATED WORK

The majority of text classification research is directed to text written in English, while little research works have been carried out on Arabic text. There are hundreds of millions of people in twenty-two countries in Asia and Africa who speak Arabic as their native language. There are more than one billion Muslims who use Arabic during their prayer and reading the Holy Quran. So, more research is needed for classifying Arabic text to satisfy the requirements of Arabic text users. Text categorization or text classification plays an important role for a lot of applications. It is concerned with assigning labels to a set of documents, where such labels are known *a priori*. Examples of such applications include, but are not limited to: classification of news, email messages and web routing. Text classification can also be used in email routing, spam filtering, automated indexing of scientific articles, searching for information on the WWW, among others [1]-[2]. Many research efforts are exerted to classify Arabic text with high accuracy. Examples of such efforts include, but are not limited to the following research studies. Laila Khreisat [3] presented a research work an classifying

[1] A. M. D. E. Hassanein is with Systems and Information Department, Engineering Division, National Research Centre (NRC), Dokki, Giza, Egypt.. Email: ahmed.diaa.hassanein@gmail.com

[2] M. Nour is with Electronic Research Institute (ERI), Cairo, Egypt. Email: mnour99@hotmail.com

Arabic text documents. The author uses the N-gram frequency statistics employing dissimilarity measures; namely, Manhattan distance and Dice's measure of similarity [3]. A comparison is made to evaluate performance using the two adopted measures. The N-gram document classification using Dice’s measure outperforms that using the Manhattan measure [3]. Majed Ismail Hussien et al. [4] presented some text classification algorithms; namely, sequential minimal optimization (SMO), Naïve Bayes (NB) and J48. The algorithms are implemented using WEKA package and operated on Arabic text. A comparative study among the adopted algorithms is carried out focusing on classification accuracy, error rate and classification time as important measurable criteria [4]. A huge number of features lead to a bad performance in terms of both accuracy and time. During the implementation work, the SMO classifier achieved the best accuracy and lowest error rate, followed by J48, then the NB classifier [4]. The SMO algorithm proved to be the fastest one, followed by NB and then J48 classifier; i.e., the J48 classifier takes the highest amount of time [4]. Fadi Thabtah et al. [5] conducted the Naïve Bayesian algorithm based on chi-square feature selection method for categorizing Arabic data. The authors presented several experimental results compared against different Arabic text categorization datasets [5]. The study concluded that feature selection often increases classification accuracy by avoiding rare or non-significant features. Riyad Al-Shalabi et al. [6] evaluated the use of K-Nearest Neighbor (KNN) to classify Arabic text. The authors used a corpus which consists of more than six-hundreds of documents that belong to six categories. They implemented a method to extract keywords based on document frequency threshold (DF) methods [6]. The work achieved about 95% micro-average precision and recall scores [6]. KNN is good with small number of training patterns, provided that there is a sufficient number of examples for each category [6]. The selection of the feature space, the training dataset and the value of K can affect the classification accuracy. Jafar Ababneh et al. [7] stated that many text categorization approaches from data mining and machine learning exist. Examples of such approaches are: decision trees, support vector machine, neural networks, statistical methods, among others. The authors presented and compared the results obtained against Arabic text collections using KNN algorithm. Three different experiments are conducted on Arabic datasets. The experimental results operated on Saudi datasets revealed that cosine similarity outperforms both Disc and Jaccard coefficients. Anshul Goyal and Rajni Mehta [8] mentioned that classification is important with broad applications. It classifies each item in a set of data into one of predefined set of classes. The authors compared between the performance of Naïve Bayes (NB) and J48 classification algorithms [8]. NB is based on probability, while J48 is based on decision tree. The comparison took place using the context of a financial institute dataset to maximize true positive rate and minimize false positive rate rather than achieving higher accuracy. The authors used classification accuracy and cost analysis as measurable criteria [8]. The results showed that the efficiency and accuracy of J48 were better than those of the NB method [8]. Adel Hamdan Mohamed et al. [9] presented a method for Arabic text categorization using support vector machine, Naïve Bayes and neural networks. The authors mentioned that several research efforts were presented for classifying English text, while unfortunately few efforts were conducted on Arabic text classification. The authors analyzed and applied the classification methods mentioned above to classify Arabic data. A comparative study was carried out using a fixed number of documents for all categories of documents in training and testing. The results showed that the support vector machine is very promising [9]. Here, we aim to apply a different approach than those applied in previous works done, where the results of each step are analyzed and evaluated. According to the results of a previous step, a next step is proposed and applied.

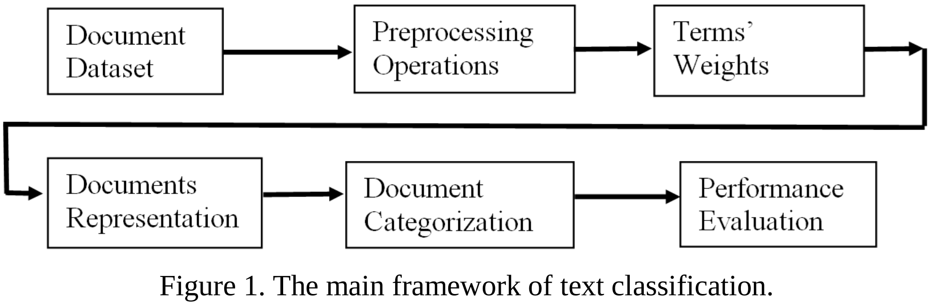

Figure 1. The main framework of text classification.

The main framework and building blocks for text or document classification are shown in Figure 1. The classification process involves several steps among which are: having a dataset and applying pre-processing operations, term weight computation, document representation and categorization and performance evaluation [10]-[15].

The dataset step is concerned with collecting the different documents that are to be classified. The pre-processing operations involve handling many themes, such as text tokenization, stop words' removal and word stemming. Document representation is important to reduce the complexity of documents and make them easy for handling. A document can be represented in a vector form using the vector space model; i.e., a document can be represented by a vector of words [12]. To perform document classification, documents are split into a training set and a testing set. The training set is used to build a model and make the system learn how to recognize different patterns of categories. The testing set is used to evaluate the system [6], [16]. Regarding the evaluation of the classification process, some measurable criteria are used. These criteria can be accuracy, precision, recall and F-measure [5]. In this paper, the same steps which are mentioned in Figure 1 are applied. The organization of this work is as follows: Section two presents an overview of the classification approaches which are used here. Section three conducts the Arabic dataset collection and the handling of some pre-processing operations. Section four introduces the proposed method for selecting the most important features for classifying the Arabic text. This step involves many themes of feature selection, such as correlation, intra-correlation, logical operations, clustering and fusion. Section five concludes the whole work and proposes possible future work.

2. Overview of Classification Approaches and Performance Evaluation

In this paper, four classifiers are adopted and investigated when classifying the dataset. The classifiers are the Naïve Bayes (abbreviated NB), Decision Tree, CART and KNN, respectively. Moreover, for the evaluation of the classification results, the measurable criteria used are accuracy, precision, recall and F-measure.

2.1 The Naïve Bayes Classifier

NB classifier works well with natural language processing (NLP) classifications. It is a supervised probabilistic algorithm that makes use of the probability theory and Bayes theorem to predict the class of a text.

NB classifier requires class conditional independence. This means that the effect of an attribute on a given class is independent of those of other attributes. If it is assumed that the training dataset $D=\{d_1,d_2......d_n\}$ contains n instances, each has a set of features and is represented as $d_i=\{x_{i1},x_{i2}......x_{in}\}$. The dataset D contains a set of classes $C=\{c_1,c_2,.....c_m\}$. Each training instance $d\in D$ has a particular class label c_i. The NB classifier predicts that an instance d belongs to a class c if and only if $P(c_i\mid d)>P(c_j\mid d)$ for $1\le j\le m, j\ne i$. The class c_i is the maximum posteriori hypothesis and it is the one for which $P(c_i\mid d)$ is maximized. The equation used is as follows [17]:

$$P(c_i\mid d)=\frac{P(d\mid c_i)\times P(c_i)}{P(d)} \tag{1}$$

where, $P(c_i\mid d)$ is the probability of document d to belong to class c. From the equation, $P(d)$ is constant for all classes, while $P(d\mid c_i)\times P(c_i)$ needs to be maximized. The class prior probabilities are calculated by $P(c_i)=\frac{|c_{i,D}|}{|n|}$, where $|c_{i,D}|$ is the number of training instances belonging to the class c_i in D and n is the number of the documents in the whole set.

2.2 The Decision Tree Classifier

The decision tree classifier is another type of supervised learning algorithms which is used in classification problems. For this algorithm, data is split into two or more homogeneous sets (or sub-populations) according to a certain splitter or differentiator in input variables. Splitting is done using several mathematical formulae. Entropy is one formula in which if p stands for success rate and q stands for failure rate, then reduction in Entropy is carried out by minimizing the formula [18]:

$$Entropy = -p\log_2(p) - q\log_2(q). \tag{2}$$

A second formula is the variance which is applied by reducing the equation [18]:

$$Variance = \frac{\sum\left(x-\overline{x}\right)^2}{n} \tag{3}$$

where, $\overline{x}$ stands for the mean of the values, x is the actual value and n is the number of values.
A third formula is Chi-square which is applied by maximizing the equation [18]:

$$Chisquare = \sqrt{\frac{(Actual - Expected)^2}{Expected}} \tag{4}$$

where, $Actual$ stands for the real class of a certain instance and $Expected$ stands for an expected class of a certain instance.

2.3 Classification and Regression Tree (CART) Classifier

CART stands for classification and regression tree which is one type of decision tree classifiers. It uses Gini method to create binary splits in a dataset. It is calculated for sub-nodes by using the sum of squares of probability for success and failure. If p stands for success rate and q stands for failure rate, the Gini formula is [19]: $\left(p^2 + q^2\right)$.

CART is important, as it deals with data using predicted and input features. CART can perform calculations and classification using both numerical and categorical parameters.

Gini index measures how well a given attribute classifies training samples into targeted classes. CART involves binary splitting of attributes, as it provides a hierarchy of univariate binary decision. The steps of CART are briefly mentioned as follows [19]: the first step is to know how the splitting attribute is selected. The second step involves setting the stopping rules and their application criteria. The third and last step is to decide on how nodes are assigned to classes.

2.4 The K-Nearest Neighbour (KNN) Classifier

The K-Nearest Neighbor belongs to the supervised learning algorithms. In this algorithm, we have a set of instances X , each having a group of features. Each instance belongs to one of a group of classes Y . The problem is to classify a new instance 'x' to one of the classes. The KNN algorithm can use several measures to define to which class or category a new instance belongs. Examples of such measures are Euclidean distance, cosine similarity, inner product similarity, among others. For the vectors of attributes (e.g. A and B), the Euclidean distance $d(A,B)$ is calculated using the following equation [20]-[22]:

$$d(A,B) = \sqrt{(a_1-b_1)^2 + (a_2-b_2)^2 + ...(a_n-b_n)^2}\ . \tag{5}$$

The k instances having the smallest distances to the new instances are grouped to choose the majority vote on the classes to which they belong. The class with the highest vote is the class of the new instance.

Some researchers use the cosine similarity in the KNN algorithm. The cosine similarity is a measure of similarity between the two vectors of n dimensions. That measure finds the cosine angle between the two vectors using the following formula [23]-[24]:

$$similarity = \cos(\theta) = \frac{A.B}{\|A\|\|B\|} \tag{6}$$

Other researchers use the inner product similarity which is known as the dot product or scalar product. It can be computed using the following formula [23]-[24]:

$$similarity = A.B \tag{7}$$

2.5 Measurable Criteria for Evaluating Performance

The performance of the proposed approach in terms of feature selection and the adopted classifiers are evaluated. To facilitate such evaluation, some measurable criteria are taken into consideration. Accuracy is one of the important themes in evaluating performance. Accuracy is defined as the ratio between the number of correctly identified documents and the total number of documents. Accuracy can be briefly expressed in terms of precision ($\Pr ec$), Recall ($\operatorname{Re} c$) and F-measure (FM), which are considered quantitative metrics. Precision ($\Pr ec$) can be defined as follows [25]-[27]:

$$\Pr ec = \frac{TP}{(TP + FP)} \tag{8}$$

Recall ($\operatorname{Re} c$) can be defined as follows [28]-[29]:

$$\operatorname{Re} c = \frac{TP}{(TP + FN)} \tag{9}$$

F-measure (FM) can be defined as follows [30]-[31]:

$$FM = \frac{2*(precision * recall)}{(precision + recall)} \tag{10}$$

where, TP is the number of documents which are correctly assigned to a certain category, FN is the number of documents which are not falsely assigned to a certain category, FP is the number of documents which are falsely assigned to a certain category and TN is the number of documents which are not correctly assigned to a certain category.

To determine the reliability of the proposed approach as well as that of the adopted classifiers, Arabic documents were taken as a test-bed. The researchers of this work selected a part of the documents in the corpus, not all of them. Very big-sized documents and very small-sized documents were avoided as explained in the next section.

3. Data Set Collection and Pre-processing Operations

We used Al-Khaleej-2004 corpus which contains more than 5000 Arabic documents. The corpus was taken from the website “https://sourceforge.net/projects/arabiccorps/”. The documents talk about daily news and are divided into four categories; namely, economy, international, local and Sport news. The average number of words and the average number of characters per word for each of the categories are calculated as shown in Table 1. The international category has the highest average number of words per document, but the lowest average number of characters per word. The economy category has the highest average number of characters per word. The lowest average number of words per document is the one calculated for the sport category. From the corpus, documents which have shooting numbers compared to the averages are rejected for our study. These documents were considered inaccurate representatives of the dataset to be selected. Including documents in the used dataset which have shooting numbers is avoided to overcome the problem of any error in the calculation of the average. The average is used in the calculation of the term weight, which is an important term in finding the other measurable criteria, such as precision and recall percentages. Equal numbers of documents from each category are chosen, so that the average numbers calculated for the whole corpora -as shown in Table 1- are still maintained for the selected ones.

Table 1. The average number of words per document and the average number of characters per word.

	Words/File	Characters/Word
Econ	461.92	6.17
Inter	561.89	5.96
Local	404.01	6.15
Sport	386.35	6.09

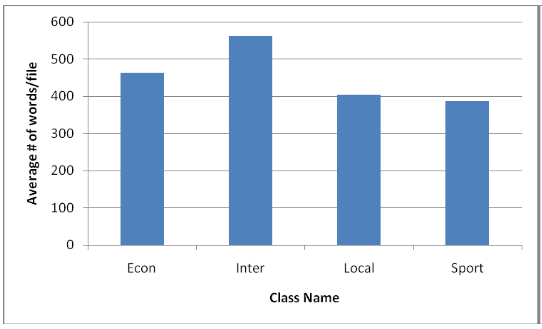

Figure 2. (a) # words/file in the dataset.

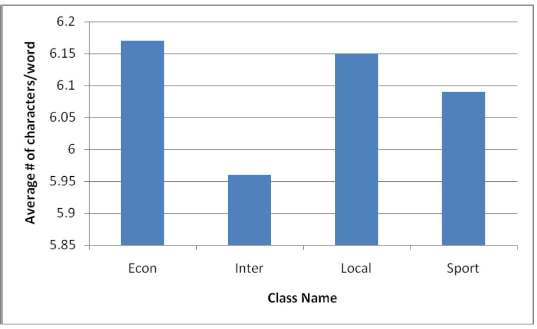

Figure 2. (b) #characters/word in the dataset.

The results in Table 1 are plotted in Figure 2 in order to clarify the differences. In this research work, WEKA and Matlab software packages are used for the calculations. WEKA is a collection of machine learning codes that can be used for data mining tasks. It contains codes for data pre-processing, classification, regression, clustering, association rules and visualization. Matlab is a commercial software package with a specialized machine learning toolbox. The creation of our features is dependent on the term or word weight. Term weight computation can be performed using many methods among which are Information gain, Relative document frequency, Chi-square, Robertson 4th formula and Robertson 1st formula [11]. A term weight is assigned to each word or feature according to its frequency in each document. If the term frequency is high and appears in few documents, that term or feature is considered important to distinguish the document contents. In this paper, the term weighting is expressed as [11]:

$$w_{i,j} = tf\left(i,j\right) \times idf\left(i,j\right) = tf\left(i,j\right) \times \log\left(\frac{n}{df(j)}\right) \tag{11}$$

where, $w_{i,j}$ is the weight of the term j in document i, $tf\left(i,j\right)$ is the occurrence of term j in document i and $idf\left(i,j\right)$ is the inverse document frequency. $df\left(j\right)$ is the number of documents which contain feature j and n is the number of all documents in the dataset.

Next, we want to represent each document d_i, where i is the document number, in an array of a number of words w_n, where n is the number of words. The problem now is to investigate the minimum number of words or features per document which are sufficient to describe each document. The Naïve Bayes (NB) classifier is initially used to test the success of classification using a different number of words to describe each document. We start by choosing a single word with the highest frequency to represent each document, so $d_i = \left(w_1\right)$, and then increase the number of words per document. We aim to find out how the increase in the number of words per document will affect the classification accuracy.

Table 2. Precision, recall and f-measure for the four categories when using different numbers of words per document. Classification was carried out using the NB classifier.

		Prec	Rec	F-M			Prec	Rec	F-M			Prec	Rec	F-M
2 words/ document	Econ	71%	16%	26%	4 words/ document	Econ	78%	78%	78%	6 words/ document	Econ	79%	67%	72%
	Inter	57%	29%	38%		Inter	96%	84%	89%		Inter	77%	96%	86%
	Local	35%	79%	48%		Local	74%	78%	76%		Local	72%	61%	66%
	Sport	46%	35%	40%		Sport	92%	94%	93%		Sport	88%	97%	92%
	Avg.	53%	41%	38%		Avg.	84%	84%	84%		Avg.	79%	79%	79%

As shown in Table 2, when the number of words used to describe a document increases from two to four words per document, the average of the four categories for the three parameters; namely, precision, recall and f-measure increases by almost 100%. However, when the number of words increases from four to six words per document, the average for the three parameters decreases by almost 5%. So, the number of words which is selected to best describe each document is four words per document. The results are plotted in Figure 3 for all the calculations performed. Table 2 shows only three examples of the performed calculations. We can find that when we use 4 words per document, the accuracy of the classification is the best.

In Table 3, the confusion matrix for the four categories is shown. Positive true rates of classification are the highest for the four categories.

Table 3. The confusion matrix for the economy, international, local and sport news categories when using four words per document.

	Econ	Inter	Local	Sport
Econ	31%	1%	6%	1%
Inter	1%	22%	3%	0%
Local	6%	0%	28%	2%
Sport	1%	0%	1%	33%

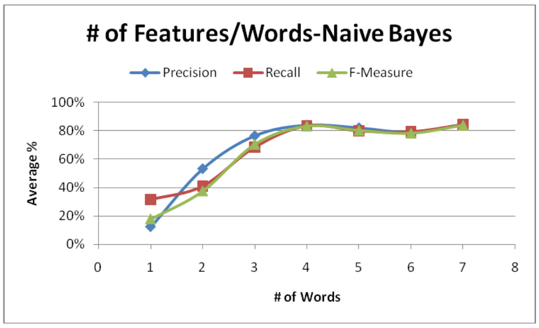

Figure 3. Measures for # words/document.

As we increase the number of words representing each document, the accuracy of the classification algorithm increases. As shown in Figure 3, we reach a point where increasing the number of words per document confuses the classification algorithm due to the repetition of words representing each document; i.e., the array of words representing each document is not any more unique for each document. Accordingly, the matrix which was fed to the WEKA software is created as follows:

1. For each document per category, the four terms with the highest term frequencies are selected to describe the content of the document.
2. For each category, we group the four words from each document for all the documents in the category and redundancy is removed.
3. For the four categories, all words that are repeated in the different categories were removed.
4. We have a matrix of 534 columns (words, attributes or features). The matrix fed to the WEKA is binary. If a word exists in a document, a one is placed in front of it; if not, a zero is placed.

4. Proposal of a Feature Selection Method

To our knowledge, 534 features are considered too large data to use in the classification problem. The time required to classify a document is large which will negatively affect the speed of calculations. Our proposal below involves many steps to reduce the number of features.

4.1 Correlation among Individual Features and Class Labels

The correlation between each of the features and the class column is calculated. Features which have the lowest correlations with the class are removed. Features are removed, so that the total number of features remaining to identify each document with its class decreases in increments of 25 features. Saying that two features have a high or low correlation with a certain class is a relative decision. A correlation value of 0.9 may be defined as high in one calculation and low in another calculation. This depends on other correlation values which we are comparing especially with the minimum and maximum values. That is why we don’t specify a threshold value to define features which are highly correlated or weakly correlated with a class. But, we chose to discard the 25 features with the lowest correlation values in every run as the threshold varies. We use the option of ten-fold calculations and the results are shown in Table 4. Figure 4 shows all calculations done, while Table 4 shows selected examples of the calculations.

In Table 4, the percentages of the precision, recall and f-measure decrease by almost 5% compared to those shown in Table 2. Precision, recall and f-measure percentages are stable as the number of attributes is decreased until reaching the knee point at 175 features. As one can see in Figure 4, we reduce the number of features from 500 to 25 and see the effect of this reduction on the accuracy of the results. When using less than 175 features in our classification, the percentages of recall and f-measure fall down. 175 features are the most efficient number to classify the documents to their corresponding classes with the highest possible accuracy. In our dataset, there exist features which are important in defining whether an instance belongs to a certain category or not. Removing features with the lowest correlations in groups of 25 features leads to the remaining of the features with have the highest correlations with the classes and which are detrimental in determining whether a document belongs to a certain class or not. Those remaining features are 175 ones. The confusion matrix when classifying documents using 175 attributes is shown in Table 5. It is shown that the highest error comes from classifying the documents under the local category. Next, we investigate why the local category is giving us the highest percentage of errors affecting all of our results later.

Table 4. Precision, recall and f-measure for the four categories when decreasing the number of attributes. The classification was done using the NB classifier.

		Prec	Rec	F-M			Prec	Rec	F-M			Prec	Rec	F-M
500 Attributes	Econ	78%	79%	79%	175 Attributes	Econ	78%	79%	79%	25 Attributes	Econ	93%	43%	59%
	Inter	94%	78%	85%		Inter	94%	78%	85%		Inter	96%	72%	82%
	Local	61%	74%	67%		Local	61%	74%	67%		Local	48%	91%	63%
	Sport	93%	86%	89%		Sport	93%	86%	89%		Sport	94%	82%	88%
	Avg.	81%	79%	80%		Avg.	81%	79%	80%		Avg.	83%	72%	73%

Table 5. The confusion matrix for the economy, international, local and sport news categories when using 175 attributes.

	Econ	Inter	Local	Sport
Econ	79%	1%	20%	0%
Inter	2%	78%	19%	1%
Local	16%	4%	74%	6%
Sport	3%	0%	11%	86%

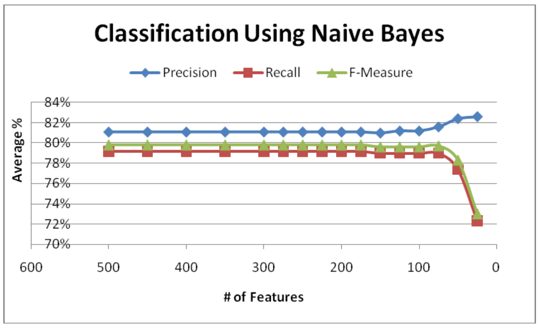

Figure 4. Measures using Naïve Bayes.

The aggregation of the four categories separately on a two-dimensional graph is examined. Two-dimensional reduction techniques are applied; namely, classical multidimensional scaling (CMDS) and a plot of the results for each two categories is shown in Figure 5. CMDS is a multivariate data analysis to explore the dissimilarity between the four classes which we have. From Figure 5, the aggregation of the points representing each category together against the three other categories is shown.

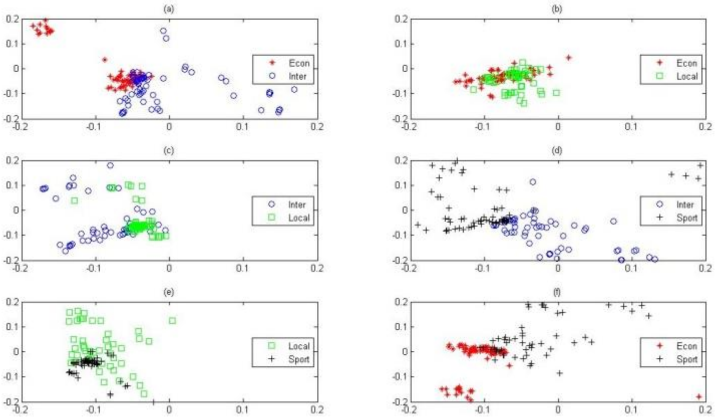

Figure 5. (a)-(f) A two-dimensional reduction in the dataset obtained for each two categories per graph. (a) Economy *versus* international, (b) Economy *vs*. local, (c) International *vs*. local, (d) International *vs*. sport, (e) Local *vs*. sport and (f) Economy *vs*. sport.

In Figure 5 (a), (d) and (f), the accumulation of the data points representing each of the economy, international and sport categories together is obvious. The separation of the data points of each of the three categories from the data points of the other two is clear as well. However, in Figure 5 (b), (c) and (e), the data points of the local category are randomly scattered through the data points of the other three categories, which makes the error in the identification of the documents belonging to the local category high. The differentiation and accordingly the classification of the documents under the local category from the other three categories have high percentage errors. This comes in accordance to what is shown in Table 4, which shows that the error is highest when classifying documents under the local category.

4.2 Operating the Adopted Classifiers on the Chosen Dataset

Different classifiers are adopted and operated on the dataset to see which kinds of classifiers can give better success rates of classification. Three classifiers; namely, decision tree, CART and KNN, are used, in addition to the Naïve Bayes (NB) classifier which has been used before.

Table 6. Precision, recall and f-measure using four different classifiers: NB, decision tree, CART and KNN classifiers.

		Prec	Rec	F-M
NB Classifier	Econ	78%	79%	79%
	Inter	94%	78%	85%
	Local	61%	74%	67%
	Sport	93%	86%	89%
	Avg.	81%	79%	80%

		Prec	Rec	F-M
Decision Tree	Econ	81%	45%	58%
	Inter	93%	59%	72%
	Local	42%	87%	57%
	Sport	93%	66%	77%
	Avg.	77%	65%	66%

		Prec	Rec	F-M
CART Classifier	Econ	81%	74%	77%
	Inter	89%	81%	85%
	Local	63%	81%	71%
	Sport	94%	84%	89%
	Avg.	82%	80%	81%

		Prec	Rec	F-M
KNN Classifier	Econ	76%	68%	72%
	Inter	84%	66%	74%
	Local	48%	77%	59%
	Sport	96%	64%	77%
	Avg.	76%	69%	70%

As shown in Table 6, the CART classifier shows the best results with 1% increase in precision, recall and f-measure compared to the NB classifier. For the CART and NB classifiers, the results of the three parameters for classifying documents under local category are first or second lowest, which is consistent with what was mentioned in Subsection 4.1. For the KNN and decision tree classifiers, precision and f-measure results for the local category are the lowest among all categories used. But, for the recall parameter, the Local category shows the highest percentage, which is inconsistent with all previous results. The calculation of the Recall parameter is inversely proportional with the false negative classification results. The recall parameter can be used in describing the success of our classification method, keeping in mind that false negative classification affects the results more than other measurable criteria. A comparison of the results is shown in Figure 6. The confusion matrix for the CART classifier is shown in Table 7.

Table 7. The confusion matrix using the CART classifier for the adopted categories.

	Econ	Inter	Local	Sport
Econ	74%	4%	21%	1%
Inter	3%	81%	15%	1%
Local	12%	4%	81%	3%
Sport	1%	2%	13%	84%

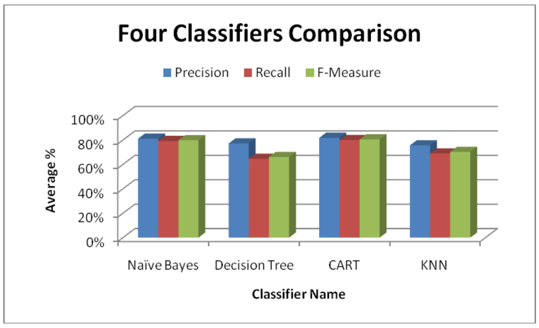

Figure 6. Average performance for the 4 classifiers used.

The classification of the documents under the local category shows better results than those for the NB classifier shown in Table 5. The error rate is lower and the success rate is higher. The CART classifier shows the best accuracy in the classification of our dataset among all classifiers used.

4.3 Intra-Correlation among Features

In this subsection, the total number of features is reduced for the whole dataset. The intra-correlation coefficients are calculated for all features. For each feature, features showing correlation values higher than a certain threshold are connected to it. Features with the highest number of connections are removed and the CART classifier is applied to see whether better classification results can be achieved or not. The threshold used here is chosen randomly to be 0.5. The number of connections is the measure to remove a feature or not. When a feature in question has a high number of connections, this means that there exist many other features which hold similar information to serve the accuracy of the classification method. We believe that removing this feature would not affect the accuracy of the classification results.

Table 8. Precision, recall and f-measure when reducing the number of features using correlation. Classification was done using the CART classifier.

174 Attributes		Prec	Rec	F-M
	Econ	81%	74%	77%
	Inter	88%	81%	84%
	Local	62%	78%	69%
	Sport	93%	84%	88%
	Avg.	81%	79%	80%

142 Attributes		Prec	Rec	F-M
	Econ	82%	73%	77%
	Inter	90%	78%	84%
	Local	59%	79%	68%
	Sport	96%	87%	91%
	Avg.	82%	79%	80%

95 Attributes		Prec	Rec	F-M
	Econ	78%	78%	78%
	Inter	94%	78%	85%
	Local	60%	74%	66%
	Sport	92%	85%	89%
	Avg.	80%	76%	77%

We aim to minimize the number of features to have percentages for the three parameters better than those shown in Table 7 or at least keep the percentages the same. As shown in Table 8, precision, recall and f-measure have the best results after removing features with the highest number of connections. As shown in Figure 7, the number of features is optimum before a drop down in the precision percentages is viewed (the knee point). The values for recall are stable until the knee point and after that, they fall down. Precision and f-measure values (81.7% and 79.9%, respectively) are the highest when the number of features is 142. Using a number of features higher than or lower than 142 decreases the accuracy of the classification method. Here, 142 is the minimum number of features which have intra-correlation values low enough so that none can substitute the existence of the others. The confusion matrix for the results of classification when using 142 features is shown in Table 9.

Table 9. The confusion matrix after removing the 17 connects using the CART classifier.

	Econ	Inter	Local	Sport
Econ	73%	2%	25%	0%
Inter	2%	78%	19%	1%
Local	13%	5%	79%	3%
Sport	0%	1%	12%	87%

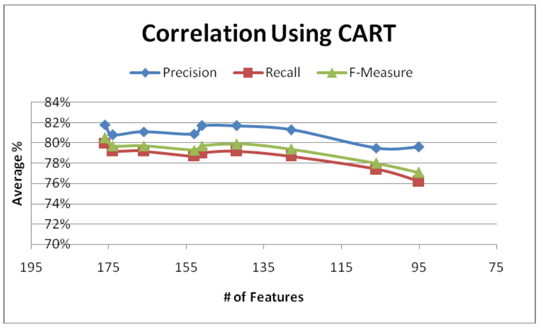

Figure 7. Measures using CART.

From Tables 7 and 9, we can see that after minimizing the number of features from 175 to 142, the accuracy achieved for the four categories is almost the same for the results of the classification percentages.

4.4 Bottom-Up Feature Fusion

Next, we apply the logical AND and OR operations to fuse the values of features together. The new features have no specific meaning, but they will reduce the total number of features. Each new feature will be the output of fusing the values of the two features into one.

Table 10. Precision, recall and f-measure when reducing the number of features using OR-binary operation. Classification is done using CART classifier.

		Prec	Rec	F-M			Prec	Rec	F-M			Prec	Rec	F-M
125 Attributes	Econ	84%	78%	81%	75 Attributes	Econ	78%	74%	76%	25 Attributes	Econ	69%	66%	67%
	Inter	94%	79%	86%		Inter	93%	87%	90%		Inter	88%	78%	83%
	Local	66%	81%	73%		Local	66%	77%	71%		Local	65%	71%	68%
	Sport	93%	92%	93%		Sport	92%	87%	89%		Sport	82%	86%	84%
	Avg.	84%	83%	83%		Avg.	82%	81%	82%		Avg.	76%	75%	76%

Inter-correlations are calculated between each two features. For the features with the highest correlation, we combine their values using logical OR. The number of features is decreased in increments of 25 features. The results of the classification are shown respectively in Table 10 and Figure 8. The best results are seen when reducing the number of features from 175 to become 125 features. The confusion matrix when using 125 features is shown in Table 11.

Table 11. Confusion matrix after reducing number of features to 125 using logical OR.

	Econ	Inter	Local	Sport
Econ	78%	1%	20%	1%
Inter	2%	79%	17%	2%
Local	12%	3%	81%	4%
Sport	0%	1%	7%	92%

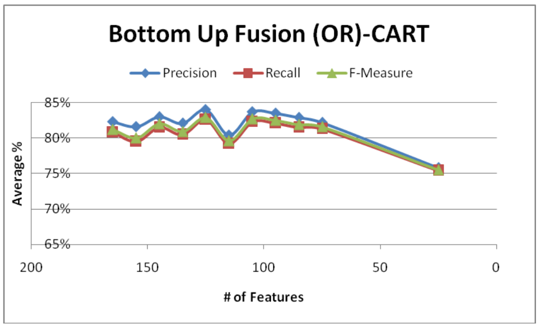

Figure 8. Measures using logical OR.

In Table 11, the results are better than those shown in Table 7. The percentage values of correct classifications for the local category are better. Next, the features are combined using the logical AND operation.

For the features with the highest correlation, we combine their values using logical AND. The number of features is decreased in increments of 25 features. The results of the classification are shown in Table 12 and Figure 9. The results are deteriorating as the number of features decreases. The confusion matrix when using 125 features is shown in Table 13.

Table 12. Precision, recall and f-measure when reducing the number of features using AND-binary operation. Classification is done using CART classifier.

		Prec	Rec	F-M			Prec	Rec	F-M			Prec	Rec	F-M
125 Attributes	Econ	59%	40%	48%	75 Attributes	Econ	100%	5%	10%	25 Attributes	Econ	24%	100%	39%
	Inter	72%	35%	47%		Inter	50%	2%	4%		Inter	0%	0%	0%
	Local	62%	34%	44%		Local	83%	5%	9%		Local	0%	0%	0%
	Sport	38%	86%	53%		Sport	27%	100%	42%		Sport	0%	0%	0%
	Avg.	57%	49%	48%		Avg.	65%	29%	17%		Avg.	6%	24%	10%

Table 13. Confusion matrix after reducing the number of features to 125 using logical AND.

	Econ	Inter	Local	Sport
Econ	40%	4%	11%	45%
Inter	4%	35%	6%	55%
Local	16%	7%	34%	43%
Sport	7%	2%	5%	86%

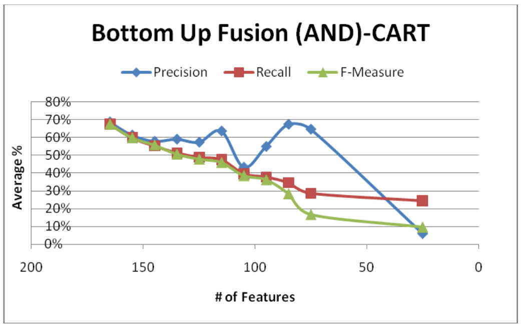

Figure 9. Measures using logical AND.

The results shown in Table 13 are the worst. The OR operation is the best method to combine different features together. Since our feature matrix is a binary one, when applying the OR logical operator on any 2 features, the output is a feature containing the information in both original features so that no information is lost. However; when applying the AND logical operator on the same 2 features, the output is a feature which only contains the shared areas so that the unshared data will be lost. That is why the classification results obtained when using the OR operator are much better than those obtained when using the AND operator.

4.5 Top-Down Feature Fusion

Next, features are combined together through unsupervised clustering. We start with all features in one cluster and then iterative methods are used to group features into two clusters. Each of the two clusters is further divided into two clusters …and so on. The results are shown in Table 14 and Figure 10.

K-means clustering is considered a partitioning algorithm. It can be used in several data mining tasks. It is considered a good algorithm to group a set of documents D into K groups or clusters. K-means clustering algorithm uses the maximum cosine similarity as a score for assigning a document to the more similar cluster centroid. The K-means algorithm is considered a proper algorithm to choose initial clusters' centroids. The document collection dataset D can be represented as $D=\left(d_1,d_2.......d_n\right)$, which can be grouped into k sets of coherent clusters. Moreover, each document d_i can be represented as a vector of weighted terms $d_i=\{w_{i1},w_{i2},.......w_{it}\}$, where t is the number of all text features in D. For more details, the reader can refer to reference [11] and [21].

Table 14. Precision,recall and f-measure when reducing the number of features. Classification is done using the CART classifier.

		Prec	Rec	F-M			Prec	Rec	F-M			Prec	Rec	F-M
25 Attributes	Econ	74%	65%	69%	100 Attributes	Econ	83%	80%	81%	150 Attributes	Econ	82%	80%	81%
	Inter	80%	78%	79%		Inter	90%	81%	85%		Inter	92%	80%	85%
	Local	59%	71%	65%		Local	67%	80%	73%		Local	63%	76%	69%
	Sport	92%	86%	89%		Sport	94%	87%	90%		Sport	92%	86%	89%
	Avg.	76%	75%	75%		Avg.	83%	82%	82%		Avg.	82%	81%	81%

It is shown from Table 14 and Figure 10 that when features are distributed into 100 clusters, the classification gives the best results. The confusion matrix for classification using 100 clusters is shown in Table 15.

Table 15. The confusion matrix after reducing the number of features to 100.

	Econ	Inter	Local	Sport
Econ	80%	3%	17%	0%
Inter	2%	81%	15%	2%
Local	12%	4%	80%	4%
Sport	2%	2%	9%	87%

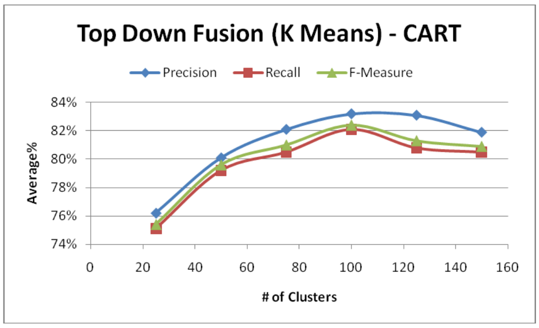

Figure 10. Measures for top-down feature fusion.

100 clusters are the best minimum number of features to classify the text under consideration. Distributing instances on clusters is grouping instances with shared values in one cluster so that at this point, fusing features under each cluster together is beneficial. But, when going further on increasing the number of clusters more than this point, the division of instances on clusters is not any more accurate and common grounds shared by instances are overstretched to the extent that two clusters may have almost similar features, which increases the classification error.

5. COMPARING PROPOSED METHOD WITH OTHERS

In this section, we compare the achieved results in this paper with the results of other feature reduction methods in two ways. The first is to use a standard feature reduction method found in WEKA and compare the accuracy of its classification results with those achieved in Table 15. The second is to compare our results with those of a previous work which used one of the state-of-the-art feature reduction techniques on the same dataset used here.

The “CfsSubsetEval” is one of the standard methods available in WEKA for feature reduction [26]. It is chosen randomly and applied to our dataset to compare the results with what we achieved in this paper. The chosen method evaluates the weight of each attribute based on its predictive ability for the class while minimizing redundancy in the final set of attributes selected. 38 attributes were selected by the “CfsSubsetEval” method [26] as the best subset of attributes that can give the highest accuracy in the classification problem. Then, the KNN classifier was applied as one of the standard available methods in WEKA for classification [26]. The results are shown in Figure 11.

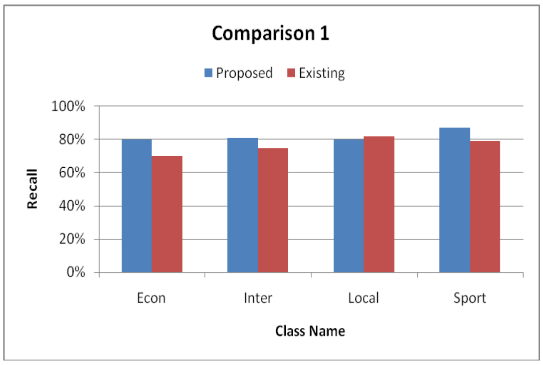

Figure 11. Comparing the achieved results with an existing standard work.

The proposed method achieved higher accuracy in the classification problem addressed, as shown in Figure 11. The proposed method produced 100 features which were needed to produce the results shown in the same figure. The accuracy of the proposed method is higher than that of the existing method for the three categories econ, inter and sport. For the local category, the proposed method still maintains a high accuracy, but slightly less than that of the existing method. As mentioned before in Section 4.1, the points representing the features of the local category are scattered through the other three categories and so, the accuracy of the identification of the documents under the local category has a higher percentage error.

In [32], one of the state-of-the-art methods of classification is used which is Multi-category Support Vector Machine (MSVM). The paper applies the method to the same dataset used here Al-Khaleej-2004. The accuracy of the results is almost equal to what we achieved with a difference of 1% more or less for the categories economy, international and sport news, as shown in Figure 12.

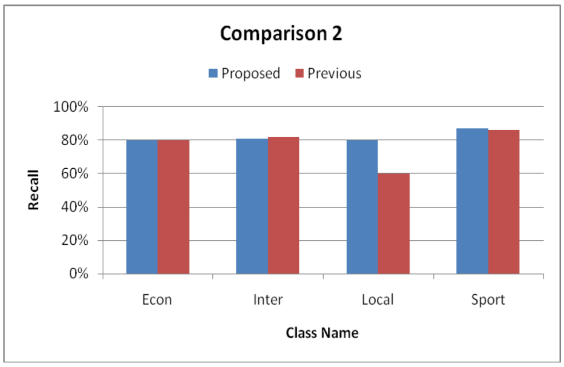

Figure 12. Comparing the achieved results with a previous work [32].

When comparing the recall values of the proposed work and the previous work for the four categories, it was noticed that the proposed work had the same, slightly better and much better values for the categories (econ and inter), sport and local, respectively, as shown in the figure. The properties of the features of the local category are discussed in Section 4.1 and we expected to have a larger error in the classification of its documents than in other categories. The proposed system succeeded in identifying the documents under the local category with almost a similar percentage error to that of the other three categories. The proposed system is resilient to problems which might exist in the used dataset than the MSVM method which was used in the previous work.

6. Concluding Remarks and Future Work

In this research work, the authors investigate and discuss the problem of Arabic text classification. Arabic documents are pre-processed by rejecting the stop words. Any document is tokenized into a set of words which are stemmed and weighted. The chosen weighted words are represented in a vector space or a feature vector. Four classifiers are operated on the chosen documents' feature vectors. The CART classifier is the best compared to the other adopted classifiers. The proposed feature selection approach improves accuracy, because it reduced the number of selected features. Precision, recall and f-measure are improved during the implementation of the steps of the proposed approach. The correlation between the individual features and the class labels, as well as the intra-correlation among the features played an important role in improving the classifier performance. Moreover, the fusion operations; either top-down or bottom-up, improve the performance of the classification process. This is clear from the values of precision, recall and f-measure, respectively. Such operations focus on selecting the most appropriate and significant features and ignoring the others. Finally, it is easy to say that the proposed feature approach can be applied on other datasets, because it is domain-independent.

Precision, recall and f-measure percentages are not as high as we would prefer to achieve. The study discussed here is based on word stemming in which the stem of all words is found and then term weighting is performed. Finding the stem of each word lacks a deeper view into the semantic relationship between the words used in each document. Including synonyms and antonyms of a word in the term weighting of the word would change the features of each document. Our proposed future work will focus on the semantics of Arabic words and how to include this deeper view into the selection of the features. New features may appear and already existing features may disappear, which can increase the accuracy of the classification method used.

References

[1] M. Suzuki, N. Yamagishi, T. Ishida, M. Goto and S. Hirasawa, "On a New Model for Automatic Text Categorization Based on Vector Space Model," Proc. of IEEE International Conference on Systems, Man and Cybernetics, pp. 3152-3159, 2010.

[2] R. Duwairi, "Arabic Text Categorization," International Arab Journal of Information Technology, vol. 4, no. 2, pp. 125-131, April 2007.

[3] L. Khreisat, "Arabic Text Classification Using N-Gram Frequency Statistics: A Comparative Study," Proc. of Conference on Data Mining (DMIN'06), pp. 78-82, 2017.

[4] M. I. Hussien, F. Olayah, M. Al-Dwan and A. Shamsan, "Arabic Text Classification Using SMO Naïve Bayesian, J48 Algorithms," International Journal of Recent Research and Applied Studies (IJRRAS), vol. 9, no. 2, pp. 306-316, November 2011.

[5] F. Thabtah, M. A. H. Eljimini, M. Zamzeer and W. M. Hadi, "Naïve Bayesian Based on Chi-Square to Categorize Arabic Data," Communication of the IBIMA, vol. 10, pp. 158-163, 2009.

[6] R. Al-Shalabi, G. Kanaan and M. Gharaibah, "Arabic Text Categorization Using KNN Algorithm,"[Online], Available: at the University of California Irvin data collections repository, http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html.

[7] J. Ababneh, O. Almomani, W. Hadi, N. K. T. El-Omari and A. Al-Ibrahim, "Vector Space Models to Classify Arabic Text," International Journal of Computer Trends and Technology (IJCIT), vol. 7, no. 4, pp. 219-223, January 2014.

[8] Anshul Goyal and Rajni Mehta, "Performance Comparison of Naïve Bayes and J48 Classification Algorithms", International Journal of Applied Engineering Research, vol. 7, no. 11, pp. 1-5, 2012.

[9] A. H. Mohamed, T. Alwada and O. Al-Momani, "Arabic Text Categorization Using Support Vector Machine, Naïve Bayes and Neural Networks," GSTF Jour. of Comput., vol. 5, no. 1, pp. 108-115, 2016.

[10] M. Labani, P. Moradi, F. Ahmadizar and M. Jalili, "A Novel Multivariate Filter Method for Feature Selection in Text Classification Problems," Eng. App. of Artificial Intell., vol. 70, pp. 25-37, 2018.

[11] L. M. Abualigah, A. T. Khader and E. S. Hanandeh, "A New Feature Selection Method to Improve the Document Clustering Using Particle Swarm Optimization Algorithm," Journal of Computer Science, vol. 25, pp. 456-466, 2018.

[12] Bhumika, S. S. Sehra and A. Nayyar, "A Review Paper on Algorithms Used for Text Classification", International Journal of Application or Innovation in Engineering and Management (IJAIEM), vol. 2, no. 3, pp. 90-99, March 2013.

[13] A. Elnahas, N. El-Fishawy, M. Nour, G. Attya and M. Tolba, "Query Expansion for Arabic Information Retrieval Model: Performance Analysis and Modification," Proc. of the Conference of Language Engineering, Cairo, December 6-7, 2017.

[14] S. A. Yousif, V. W. Samawi, I. Elkaban 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.

[15] M. M. Hijazi, A. M. Zaki and A. R. Ismail, "Arabic Text Classification: Review Study," Journal of Engineering and Applied Sciences, vol. 11, no. 3, pp. 528-536, 2016.

[16] S. Osama and M. Nour, "Feature Selection Methods for Predicting the Popularity of Online News: Comparative Study and a Proposed Method," Journal of Theoretical and Applied Information Technology, vol. 96, no. 19, pp. 6969-6980, October 15, 2018.

[17] D. Md. Farid, Li Zhang, C. M. Rahman, M. A. Hossain and R. Strachan, "Hybrid Decision Tree and Naïve Bayes Classification for Multi-Class Classifications Tasks," Journal of Expert Systems with Applications, vol. 41, pp. 1937-1946, 2014.

[18] A. Brunello, E. Marzano, A. Montanari and G. Sciavicco, "J48SS: A Novel Decision Tree Approach for the Handling of Sequential and Time Series Data," Computers Jour., vol. 8, no. 21, pp. 1-28, 2019.

[19] E. Venkatesan and T. Velmurugan, "Performance Analysis of Decision Tree Algorithms for Breast Cancer Classification," Indian Journal of Science and Technology, vol. 8, no. 2, pp. 1-8, November 2015.

[20] Z. Elberrichi and K. Abidi, "Arabic Text Categorization: A Comparative Study of Different Representation Modes," International Arab Journal of Information Technology, vol. 9, no. 5, pp. 465-470, September 2012.

[21] M. Allahyari, S. Pouriyeh, M. Assefi, S. Safaei, E. Trippe and J. Gutierrez, "A Brief Survey of Text Mining: Classification, Clustering and Extraction Techniques, " KDD Bigdas, Halifax, Canada, pp. 1-13, July 2017.

[22] P. Kumbhar and M. Mali, "A Survey on Feature Selection Techniques and Classification Algorithms for Efficient Text Classification," Int. Jour. of Science and Research, vol. 5, no. 5, pp. 1267-1275, 2016.

[23] M. Abbas and K. Smaili, "Comparison of Topic Identification Methods for Arabic Language," Proc. of the International Conference of Recent Advances in Natural Language Processing (RANLP'05), Borovets, Bulgary, pp. 14-17, September 21-23, 2005.

[24] I. Rouby, M. Badawy, M. Nour and N. Hegazi, "Performance Evaluation of an Adopted Sentiment Analysis Model for Arabic Comments from the Facebook," Journal of Theoretical and Applied Information Technology, vol. 96, no. 21, pp. 7098-7112, November 15, 2018.

[25] N. Bhargava, G. Sharma, R. Bhargava and M. Mathuria, "Decision Tree Analysis on J48 Algorithm for Data Mining," International Journal of Advanced Research in Computer Science and Software Engineering, vol. 3, no. 6, pp. 1114-1119, June 2013.

[26] V. P. Bresfelean, "Analysis and Predictions on Students' Behavior Using Decision Trees in WEKA Environment," Proceedings of the 29th IEEE International Conference on Information Technology Interfaces, Croatia, June 25-28, 2007.

[27] T. R. Patil and S. S. Sherekar, "Performance Analysis of Naïve Bayes and J48 Classification Algorithms for Data Classification," International Journal of Computer Science and Applications, vol. 6, no. 2, pp. 256-261, April 2013.

[28] M. F. Zaiyadi and B. Baharudin, "A Proposed Hybrid Approach for Feature Selection in Text Document Categorization," International Journal of Computer and Information Engineering, vol. 4, no. 12, pp. 1799-1803, 2010.

[29] S. Francisca Rosario and K. Thangadurai, "RELIEF: Feature Selection Approach," International Journal of Innovative Research and Development, vol. 4, no. 11, pp. 218-224, October 2015.

[30] R. P. Durgabai, "Feature Selection Using Relief Algorithm," International Journal of Advanced Research in Computer and Communication Engineering, vol. 3, no. 10, pp. 8215-8218, October 2014.

[31] U. G. Mangai, S. Samanta, S. Das and P. R. Chowdhury, "A Survey of Decision Fusion and Feature Fusion Strategies for Pattern Classification," IETE Technical Review, vol. 27, no. 4, pp. 293-307, 2010.

[32] M. Abbas, K. Smaïli, and D. Berkani, "Multi-Category Support Vector Machines for Identifying Arabic Topics," Research in Computing Science, vol. 41, pp. 217-226, 2009.

ملخص البحث:

يلعب تصنيف النصوص باللغة العربية دوراً مهماً في العديد من التطبيقات. ويهدف تصنيف النصوص الى تخصيص أصناف معرّفة مسبقاً للوثائق النصية؛ إذ إن النصوص العربية غير المهيكلة ربما تكون سهلة المعالجة من جانب البشر، لكن فهمها وتفسيرها من جانب الآلة يكونان أكثر صعوبة. لذا، فإنه قبل تصنيف النصوص العربية، لا بد من القيام ببعض العمليات التي تدخل في باب المعالجة المسبقة.

يقدم هذا العمل البحثي أنموذجاً لانتقاء السِّمات من نصوص أو وثائق باللغة العربية. وهنا تستخدم كلمة (نصّ) وكلمة (وثيقة) بالمعنى ذاته. وقد تم استقاء النصوص لهذا العمل من (مجموعة الخليج)-2004. وتتضمن هذه المجموعة آلاف الوثائق التي تتناول أخباراً في مجالات مختلفة؛ مثل الأخبار الاقتصادية، والأخبار العالمية، والأخبار المحلية، وأخبار الرياضة. وقد تم إجراء عدد من عمليات المعالجة المسبقة من أجل استخلاص المفردات عالية الوزن التي تصف محتوى الوثيقة على النحو الأفضل. ويتضمن الأنموذج المقترح العديد من الخطوات من أجل تعريف السمات الأكثر ملاءمة. وبعد تحديد العدد الأولي من السمات، بناءً على الكلمات الموزونة، تبدأ خطوات الأنموذج المقترح. الخطوة الأولى مبنية على حساب الارتباط بين كل سمة من السمات والصنف الأول. وبناءً على قيمة عتبة محددة، يجري انتقاء السمات الأعلى ارتباطاً؛ الأمر الذي يقود الى تقليل عدد السمات المنتقاة. ثم يجري التقليل من عدد السمات مرّة أخرى عبر حساب الارتباط بين السمات الناتجة، ويكون ذلك في الخطوة الثانية. ومن خلال القيام ببعض العمليات المنطقية، يتم في الخطوة الثالثة انتقاء أفضل السمات من بين تلك التي نتجت من الخطوة الثانية، وذلك بناءً على عمليات (AND) و(OR) من أجل دمج بعض السمات تأسيساً على بِنْيتها وطبيعتها ودلالتها. وينجم عن ذلك تقليل آخر من عدد السمات. أما الخطوة الرابعة، فتقوم على فكرة (عَنْقَدة) النص أو الوثيقة، بحيث يتم وضع السمات التي نتجت من الخطوة الثالثة في مجموعة واحدة ومن ثم إجراء عمليات تكرارية تؤدي الى وضع السمات في مجموعتين، ليصار بعد ذلك الى تجزئة كل مجموعة من السمات الى مجموعتين ... وتستمر التجزئة الى أن تصبح محتويات المجموعات ثابتة لا تتغير. ويجري دمج محتويات مجموعات السمات باستخدام قاعدة جيب التمام؛ الأمر الذي يقلل العدد الإجمالي للسمات.

يستخدم هذا العمل أربعة مصنِّفات هي: (KNN, CART, DT, NB) للمقارنة بينها من حيث عدد السمات الناجم عن استخدام كل منها. وتأخذ الدراسة المقارنة بعين الاعتبار عدداً من المعايير، وهي: (F-M, R, P)، إضافة الى دقة التصنيف). وقد تم تطبيق هذا الأنموذج باستخدام حُزم البرمجة: ويكا (WEKA)، وماتلاب (MATLAB). وتشير النتائج التي تم الحصول عليها الى أنّ المصنف (CART) كان الأفضل من حيث الأداء، بينما كان الأسوأ أداءً المصنف (KNN).