RUJCST
جامعة الرازي
Al-Razi University
كلية الحاسوب وتقنية المعلومات
مجلة جامعة الرازي لعلوم الحاسوب وتقنية المعلومات
Al-Razi University Journal of Computer Science and Technology
علمية محكمة تصدر عن كلية الحاسوب وتقنية المعلومات - جامعة الرازي

Comparative Study of Five Summarization Approaches for Arabic Documents Using Text Classification

1st Khaled Alwesabi
IT. Al-Razi university.
Al-Razi university
Sana'a, Yemen
koalwesabi5022@gmail.com

2nd Abdullah Ayedh
IT. YemenSoft Company
YemenSoft Company
Yiwu- China
abuhany20@hotmail.com

3rd Yahya Al-Ashmoery
Department of Information Technology, Al-Razi University,
Department of Mathematics & Computer, Faculty of Science
Sana'a University, Yemen
Yah.AlAshmoery@su.edu.ye

4th Hisham haider
AI. Al-Razi university.
Al-Razi university
Sana'a, Yemen
hesham_haider@yahoo.com

***Abstract*— Text documents are continuously increasing every day so that long time will be spent to deal with all those documents, in addition, text summarizing reduces the required time and efforts needed to explore and identify the most relevant and salient parts of the body of text. Moreover, text classification helped in facilitating access to study required fields quickly. We introduced five methods for summarizing Arabic documents to get best method with high efficiency and accuracy. The summarization methods used in this paper are LexRank, Degree Centrality, Continuous LexRank, Centroid Based and Lakhas, while classification method used is supported factor machine (SVM). We examined whether the use of document classification to evaluate what the best method for Arabic document summarization. In other words, we get best approach to summarize document through classification. The summarizer performance is evaluated in terms of the efficiency and accuracy by precision, recall, and the execution time. Finally, a comparison between the summarization methods using the classification is conducted. Experimental results show that the summarization by Centroid Based method then classification can achieve an accuracy by more than 96.96% in a time of 03:42 minutes comparing with other summarization methods. Classification efficiency is also significantly improved when the classification is based on summaries especially when Centroid Based method has been used, rather than full-length documents. In addition, memory space required and run time for classifying summarized documents are less than the memory and time needed for classifying full documents.**

Keywords*— *Text summarization, Text Classification, Lakhas method, Support victor machine (SVM), Centroid Based method

I. INTRODUCTION

Sifting of thousands of heterogeneous data sources is required during data recovery over the Web. Vital sources of data include a growing variety of non-traditional, semi- or unstructured collections such Web sites, FTP archives, etc. rather than standard databases with organized information and inquiries. There is a need for new methods of summarizing, locating, and choosing collections relevant to a user's enquiry as the quantity and changeability of sources increase. Processing text is one of the major fields of data engineering management and information retrieval. Important work has been undertaken in the field since early the history of Information Technology. Text management includes subjects as document summarization and document classification.

Text summarization [1] is a condensed version of the original document or several documents. This condensed version contains the most important relevant information in context found in the source document [2]. With summaries, we can make effective decisions and get useful information in less time. The require for content summarization comes from the huge sum of electronic reports and the require for sparing handling time and make viable choices. Text classification has become one of the key techniques for handling and organizing text data. Text Classification necessity comes from the large amount of electronic documents [3] into a number of predefined categories. Machine learning algorithms such as decision trees [4], neural network [5] and Bayesian classifiers [6] have been used for such a purpose. The classification accuracy is affected by the content of documents and the classification technique being used. In this paper, we examine whether the use of document summarization can result in

better classifications and what are the best of these methods through classification.

The main motivation came from using text classification before and after summarization is the huge number of features and diversity or terms representing documents. If these terms can be reduced without affecting the value or content of documents, then memory space and classification processing time will be saved. Each document may belong to more than one category. Currently, DC is widely used in different domains, such as mail spam filtering, article indexing, Web searching and Web page categorization [7, 8]. These applications are increasingly becoming important in the information-oriented society at present.

In this paper, we'll use the classification of text twice, use it before summarization (full documents) and after summarization. then measured the accuracy of classifying before and after applying the summarization. We show that Centroid Based summarizer significantly outperformed the other summarization techniques for all classification experiments and provides significant improvement in the accuracy of document classification, also using text summarization techniques for preprocessing in text classification is a viable and effective technique.

The rest of the paper is organized as follows. In Section 2, we present the related works on web classification and summarization. Some studies that rely on document classification, by summarization, then we present our proposed approach in Section 3. In Section 4, we display summary about on different summarization methods, also we will introduce classification method used in Section 5, while in Section 6, we proved by experimental results comparing five summarization methods by classification. The final section of this paper concludes with future work in the area, underlining the feasibility of our results.

II. RELATED WORKS

Arabic is the native language of more than 330 million speakers [9] living in an important region with huge oil reserves crucial to the world economy, in an area extending from the Arabian/Persian Gulf in the East to the Atlantic Ocean in the West. Unlike the Latin script, orientation of writing in Arabic is from right to left; the Arabic alphabet consists of 28 letters. Arabic words have two genders; feminine and masculine, three number cases; singular, dual, and plural, and three grammatical cases; nominative, accusative, and genitive. Words in Arabic language are classified into three main parts of speech, nouns (including adjectives and adverbs), verbs, and particles [10].

Studies that use text summarization to improve text classification are limited. Recently, these studies effectively begin researched, special about web-page summarization [11-16]. Ocelot is a system for summarizing Web pages using probabilistic models to generate the ‘‘gist’’ of a Web page [12]. The models used are automatically obtained from a collection of human-summarized Web pages. [11]. Proposed a new webpage classification algorithm based on web summarization for improving the accuracy. According to experimental findings, summarization-based classification is 8.8% better than pure-text-based classification. The study in [15] suggested using summarization techniques to reduce noise and boost Web page classification performance. Furthermore, classification algorithms (NB or SVM) can outperform pure-text-based classification algorithms by more than 5.0% when they are supported by any summarizing approach. Moreover, an ensemble method was created to combine the various summarizing algorithms. In comparison to pure-text based approaches, the ensemble summarization method improves performance by more than 12.0%. The authors in [13] presented five methods for summarizing portions of Web pages on handheld devices, where the core algorithm is to compute the importance of the words using TF/IDF measures and to select important sentences using Luhn's classical methods [14, 16]. These methods also took advantage of the effect of context in Web-page summarization, which is the information extracted from the content of all the documents related to a page. Evidence suggests that summaries that consider context information are typically more pertinent than those that are just based on the subject document.

Some studies have been conducted to enhance document categorization by summarization [17-20].

A Study in [17] presented an assessment of the most 15 widely used methods for automatic text summarization from the text classification perspective. A naive Bayes classifier was used showing that some of the methods tested are better suited for such a task.

In [18], summarizing materials using an automatic text summarizer has been done. Before and after applying the summarizing, two classification techniques were applied to categorize Arabic documents, and the classification accuracy of the complete documents and the summarized documents were compared. The classification accuracy obtained by categorizing complete materials is comparable to that obtained by categorizing summarized documents. However, the memory and processing time needed for classifying summary materials are smaller than those for classifying complete documents.

The authors introduced a hybrid text classification model in [19]. A Support Vector Machine (SVM) classifier is used to categorize unlabeled texts. Instead of using the source papers, classification rules are produced using summaries of the training materials. A document's summary is produced using latent semantic analysis (LSA). The ideal number of phrases for summary generation is determined via empirical LSA analyses. Precision, recall, and F1 ratings are used to assess how well text summarizing using LSA and text classification using summaries perform.

The aim of the work in [20] was to detect whether classification of documents can help with guided document summarization. This method takes into account a number of classes into which a text could be classified and a novel summary technique designed to extract summaries in accordance with the classification results. The system performs admirably when measured against a variety of supervised and unsupervised alternatives.

In this paper, we focus on the advantages of some existing classification algorithms to get best summarization method from through classification. An Effective summarization algorithm is proposed, with some features related to classification are integrated into some existing summarization methods. The aim is to compare the five ways of summarization through the use of the classification in order to obtain the best way to summarize that helps in text classification.

III. THE PROPOSED APPROACH

The corpus sets that presented in this research is divided by 3: 1 training/test. The corpus is parsed into one or more categories. The task of text classification consists of finding the most probable category for a new unseen document, based on features extracted from training examples. The classification process is often performed using information drawn from entire documents, and this may sometimes result in noisy features. To lessen this effect, we propose to feed the text classifier with summaries rather than entire texts, with the goal of removing the less important, noisy sections of a document prior to classification..

In order to test the effectiveness of summarization for text classification, several experiments are conducted. With knowing that the main aim of this paper is to identify the best summarization method to improve classification. The main parts of the paper proposal are described below:

- feed the text classifier with summaries.
- Test the text classification depending on the original text.
- Test the text classification on the systems generated summaries in order to find out whether classification can help to determine the most effective method to summarize texts at all.
- We compare test first result ‘‘text classification original’’ with test second result "Classification of texts resulting from each summarization process (five methods summarization)" including: Lakhas algorithm, LexRank, Continuous LexRank, Centroid and the graph-based method (Degree Centrality).
- The summarization methods are evaluated based on experiments, as we studied performance of best summarization method and that outperformed the other summarization techniques for all classification experiments, it also provides significant improvement in the accuracy of document classification. Figure 1 illustrates the classification process based on extractive summarization.

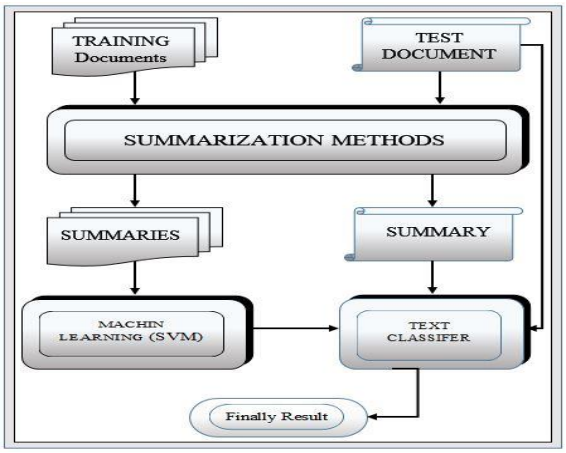


Fig. 1. Architecture of Proposed System

IV. DOCUMENT SUMMARIZATION

Summarizing reduces the required time and effort needed to explore and identify the most relevant and salient parts of a body of text. In addition, although the field of automatic summarization is over 50 years old [16, 21], illuminating of Arabic automatic summarization is more recent and still not on par with the research on English and other European languages. Work on Arabic summarization started less than 10 years ago, [22, 23].

In particular, we will consider five different methods for conducting the documents summarization. The first method corresponds to an adaptation of Lakhas summarization technique. The second method corresponds to using LaxRank on documents for summarization. The third method corresponds to finding the important content body by Continuous LaxRank summarization. The fourth one is based on graph analysis using degree centrality. Finally, we will use centroid-based summarization, the sentences that contain more words from the centroid of the cluster are considered as central.

In our study, we focus on text summarization with the benefit of the text classification in the comparison between of the methods known in the texts summarization. Methods Summarization that covered this study.

A. Sakhr Summarizer: (Lakhas Method)

A Study in[23], used a weighted linear combination of these features (which is often the case) to score sentences as shown in equation (1).

The architecture of the system as illustrated in Figure 1 consists of following parts:

- Sentence segmentation.
- Word segmentation (tokenization).
- Normalization (example أ, إ and آ were replaced by ا, ى was replaced by ي and ه was replaced by ة,….).
- Stop words removal.
- lemmatization (simple prefix and suffix removal).
- Frequency computation.
- Indicative expressions (Cue words).
- Weight computation of each sentence S is obtained by combining the value of 4 scores:

$$Sc = a_1 Sc_{lead} + a_2 Sc_{title} + a_3 Sc_{cue} + a_4 Sc_{tf.idf} \qquad (1)$$

Where

Sc_{lead} is 2 if the sentence is first one and 1 otherwise

$$\sum_{w \in S} a(w).tf(w) \qquad (2)$$

Where

$$\begin{cases} a(w) \;\; is\ 2\ if\ word\ w\ appears\ in\ the\ title\ of\ the\ article\ and \\ 0\ otherwise \\ tf(w) \qquad is\ the\ frquency\ of\ w\ in\ the\ sentence \end{cases}$$

$$Sc_{cue} = \sum_{w \in S} c(w).tf(w) \quad (3)$$

Where

$$\begin{cases} c(w) \ \ is\ 1\ when\ w\ appears\ in\ the\ list\ of\ indicative\ words \\ \qquad\qquad \text{and 0 otherwise} \\ tf(w) \qquad is\ the\ frquency\ of\ w\ in\ the\ sentence \end{cases}$$

$$Sc_{tf.idf} = \frac{1}{|S|} \sum_{w \in S} \frac{tf(w) - 1}{tf(w)} log \frac{DN}{df(w)} \quad (4)$$

Where

$$\begin{cases} tf(w) \ \ is\ the\ frequency\ of\ w\ in\ S. \\ DN\ is\ the\ total\ number\ of\ documents\ in\ the\ corpus. \\ df(w)\ is\ the\ number\ of\ documents\ in\ wich\ w\ occurs \end{cases}$$

For DUC2004, we set all a_i to 1 but we intend to experiment with different values. Look [23].

- **Sentence extraction and compaction**. The above steps are sufficient for short summaries of a few sentences. Figure 2 gives a functional view of Lakhas in terms of the modules that we now briefly describe:

Fig. 2. Modules of Lakhas used in the DUC competition

B. Degree Centrality Summarizer:

The degree centrality of a sentence is the degree of the corresponding node in the similarity graph, which shows the effect of cosine threshold selection. Too high thresholds may lead to lose many of the similarity weights in a set of documents, on the other hand, too low thresholds may lead to weak similarity weights into consideration [24, 25].

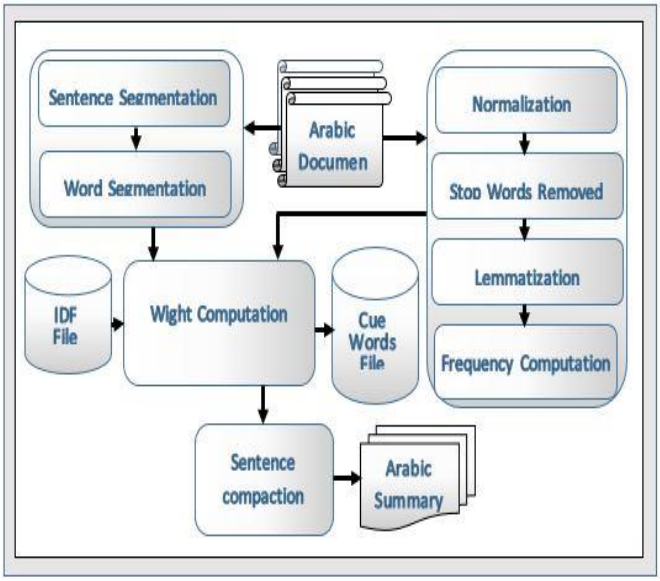
In the degree centrality methods, it is necessary to formulate the problem as follows:

- Represent each sentence by a vector
- Denote each sentence as the node of a graph
- Cosine similarity determines the edges between nodes (look figure 3). similarity measure (look figure 4) is used to compute the similarity between two sentences as follows:

$$idf - modified - cosine(x,y) = \frac{\sum_{w \in x,y} tf_{w,x} tf_{w,y} (idf_w)^2}{\sqrt{\sum_{x_i \in x} \left(tf_{x_i,x} idf_{x_i}\right)^2} \times \sqrt{\sum_{y_i \in y} \left(tf_{y_i,y} idf_{y_i}\right)^2}} \quad (5)$$

Where: $tf_{w,x}$ is the frequency of the word w in the sentence s and idf_w is the inverse document frequency A cosine similarity matrix is computed and used for a cluster representation, where each item in the matrix represents the similarity between the corresponding sentences pair [26, 27].

- Since we are interested in significant similarities, we can eliminate some low values in this matrix by defining a threshold.
- Compute the degree of each sentence



- Pick the nodes (sentences) with high degrees

Fig. 3. Weighted cosine similarity graph

Fig. 4. Similarity graphs that correspond to thresholds 0.1, 0.2, and 0.3.

C. *Centroid-Based Summarizer:*

Extractive summarization works by selecting a subset of the sentences in the original text to form the summary. Thus, can determine the most central sentences in a (multi-document) cluster that give the necessary and sufficient amount of information related to the main theme of the cluster.

In centroid-based summarization [27, 28], the sentences that contain more words from the centroid of the cluster are considered as central. This is a measure of how close the sentence is to the centroid of the cluster.

Centroid-based summarization has given promising results in the past, and it has resulted in the first web-based multi-document summarization system[1] [29]. The sentences that contain more words from the centroid of the cluster are called central as Algorithm.

Fig. 5. Example of a figure caption. (*figure caption*)

D. *Lexrank Summarizer:*

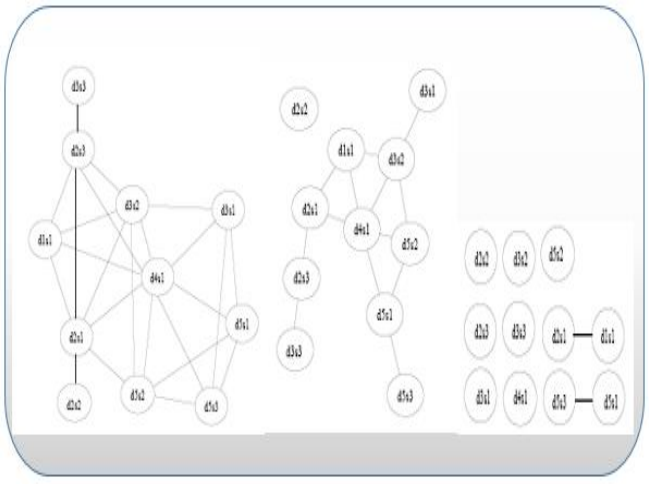
(Algorithm Centroid-Based Summarizer). (Computing Centroid Scores Algorithm)
Input: An Array S of n sentences, cosine threshold t

Output: An array C of Centroid scores

```
Hashtable centroid = new Hashtable();
float[] C =new float[]();

/* Compute tf_idf scores for each word*/
for( int i =0;i< n;i++)
{
foreach(string  w in S[i] )
{
centroid[w].tfidf = centroid[w].tfidf + idf(w);
}//end of foreach
}//end of for

/* Construct the centroid of the cluster*/
/* by taking the words that are above the threshold*/
foreach(string w in centroid )
{
if(centroid[w].tfidf > t )
centroid[w].centroid =  centroid[w].tfidf;
else
centroid[w].centroid =0;

}//end foreach

/* compute the score for each sentences*/
for(int i = 0;i<  n ;i++)
{
C[i] = 0;

foreach(string w in  S[i])
{
C[i] = C[i] + centroid[w].centroid;
}//end foreach
}//end for

return C;
```

The LexRank algorithm proposed by Gunes Erkan and Dragomir Radev from University of Michigan refers to the method used to calculate the weight of sentences under graphic expression of sentences. They think if one sentence bears much similarity with other sentences, the sentence would be fairly important [27].

A straightforward way of formulating idea Lexrank is to



consider every node having a centrality value and distributing this centrality to its neighbors. Centrality vector p which will give a Lexrank (lexical PageRank) of each sentence (similar to page rank) defined by the equation:

[1] http://www.newsinessence.com

$$p(u) = \sum_{v \in adj[u]} \frac{p(v)}{deg(v)} \quad (6)$$

where p(u)is the centrality of node u, adj [u] is the set of nodes that are adjacent to u, and deg (υ) is the degree of the node υ. Equivalently, we can write Equation 3 in the matrix notation as

$$P = B^T P \quad (7)$$

Or

$$P^T B = P^T \quad (8)$$

where the matrix B is obtained from the adjacency matrix of the similarity graph by dividing each element by the corresponding row sum:

$$B(i,j) = \frac{A(i,j)}{\sum_k A(i,k)} \quad (9)$$

We assign a uniform probability for jumping to any node in the graph, which is known as PageRank.

$$p(u) = \frac{d}{N} + (1-d) \sum_{v \in adj[u]} \frac{p(v)}{deg(v)} \quad (10)$$

where N is the total number of nodes in the graph, and d is a "damping factor", which is typically chosen in the interval [0.1, 0.2] [30]. Equivalently, we can write the equation in the matrix notation as,

$$P = [DU + (1-d)B]^T P \quad (11)$$

Here, U indicates that all elements there equal to matrix of 1/N. [27].

E. Continuous LexRank Summarizer:

In order to compute Degree centrality constructed the similarity graphs and LexRank are unweighted. Because the binary Individualization we perform on the cosine matrix using an appropriate threshold. As in all Individualization operations, it means information might possibly go missing. Similarity link intensity could be used to improve the LexRank.

If we use the cosine values directly to construct the similarity graph, we usually have a much denser but weighted graph. We can normalize the row totals of the corresponding transition matrix to establish a random matrix. The resultant equation is a modified version of LexRank for weighted graphs:

$$p(u) = \frac{d}{N} + (1-d) \sum_{v \in adj[u]} \frac{idf-modified-cosine(u,v)}{idf-modified-cosine(v)} p(v) \quad (12)$$

From the above method, while computing LexRank for a sentence, we multiply the LexRank values of the linking sentences by the weights of the links. The weight realized the regulation according to sum of lines and the paper adds the damping factor d into the convergence of the method [27, 31].

V. DOCUMENT CLASSIFICATION

Three ways, including unsupervised, supervised, and semi-supervised methods, can be used to classify the documents. The classification SVM model is used to comprise the chosen features and the accompanying weights in each document of the training dataset. Testing data will be utilized in order to evaluate the classification model that was built throughout the training procedure. Recently, several methods and algorithms for automatically classifying documents have been presented. Automatic document classification typically uses supervised learning techniques [32] such as support vector machine (SVM) [37, 38], naïve bayes (NB) [33, 34] and, k-nearest neighbor (KNN) [35, 36] etc. where pre-defined category labels are assigned to documents based on the likelihood suggested by a training set of labelled documents [32].

In this work, SVM classifier with linear kernel was applied. A brief overview on the SVM classification algorithm is presented in the following.

A. Support Vector Machine (SVM)

Due to the high performance, Support Vector Machine has recently become a popular algorithm. SVM is also able to handle documents with a high dimensional input space and remove the majority of the superfluous features [39].

A common machine learning method is support vector machine. The input samples in N-dimensional space are translated onto a higher dimensional space according to the structural risk reduction concept, and a maximal separation hyperplane is then detected [40].

A support vector machine creates a hyperplane or set of hyperplanes in a high dimensional space that may be implemented to tasks like regression and classification.

The SVM's steps are listed in brief as follow:

- Use a kernel function to map the data to a pre-defined and high-dimensional space.
- Find the hyperplane where the margin between the two classes is at its greatest.
- locate the hyperplane that maximizes the margin and minimizes the (weighted average of the) misclassifications (in case the data cannot be separated).

The usage of SVMs can be applicable to both linear and nonlinear problems. The original training data is transformed into a higher dimension via a nonlinear mapping. It looks for the linear optimal separation hyperplane with the new dimension. Data from two classes can always be separated by

a hyperplane with the right nonlinear mapping to a high enough dimension. SVMs finds this hyperplane using support vectors and margins.

Consider a training set of labeled instances $x_i \in R^n, i = 1....L$, belong to a set of categories $y_i \in |-1, 1|$.

Figure 6: is an example of an optimal hyper plane for separating two classes [41].

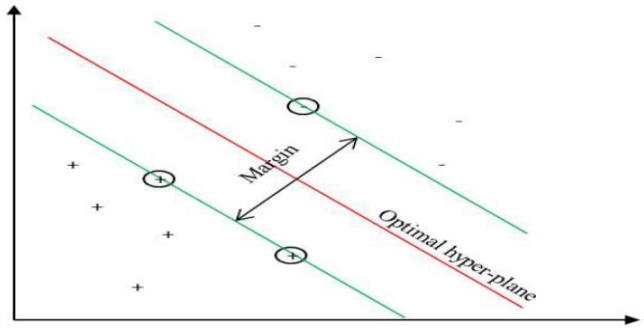


Fig. 6. An example of an optimal hyper plane for separating two classes

From Figure (6), SVM builds the classification model on the training data using a linear separating function to classify unseen instances [42].

For linearly separable vectors, the kernel function is simple. It takes the form:

$$f(x) = W.X + b \qquad (13)$$

W is called weight vector for optimal hyper-plane and b is known as the bias. The class of X (test instance) can be found using the following linear decision function [40, 42]:

$$y = sign(f(x)) \qquad (14)$$

The optimal separating hyper plane is the one that has the largest margin. The distance between nearest vectors to the hyper plane is maximal. The distance is given by:

$$min_{x_i,\ y_i=+1} \frac{(w+X)+b}{|w|} - max_{x_i, y_i=-1} \frac{(w+X)+b}{|w|} = \frac{2}{|w|} \qquad (15)$$

The hyper plane which minimizes w is considered as the optimal hyper plane [41].

$$\frac{1}{2}\|w\|^2$$

Using Lagrangian formula, the maximal margin hyper plane can be rewritten as:

$$f(x) = \sum_{i=1}^{n} a_i y_i k(x, x_i) + b \qquad (16)$$

Where y_i is the class label of support vector x_i, k (x, x_i) is the kernel function, x is a test vector, n is the number of support vectors, ai is a Lagrange multiplier for each training vector (vectors for which $a_i > 0$ are called support vectors), and b is a scalar (a numeric parameter). For the text classification problem, y_i and y_i represents the i_{th}ith document and the class of that document (e.g. sport, science, religion, etc.) in the training set, respectively [40, 41].

For non- linear data, various kernel functions can be utilized with SVM. The most popular kernel functions are represented as follow [40, 42]:

- Polynomial kernel:

$$K(x_i, x_j) = (\gamma x_i \cdot T \cdot x_j + r)^d, \gamma > 0 \qquad (17)$$

- RBF kernel:

$$K(x_i, x_j) = \exp(-\gamma \| x_i - x_j \|^2), \gamma > 0 \qquad (18)$$

- Sigmoid kernel:

$$K(x_i, x_j) = tan(\gamma x_i \cdot T \cdot x_j + r) \qquad (19)$$

Here, γ, r and d are kernel parameters.

If the number of features is very large as in the document classification example, the linear kernel function is the right choice and there is no need to map the data [42].

Because the complexity of a trained classifier is determined by the number of support vectors rather than the dimensionality of the data, SVMs are effective on high-dimensional data. Support vectors are the essential training examples because they are the closest to the decision boundary; if all other training examples are eliminated and the training is repeated, the same separating hyperplane would be discovered. Independent of the dimensionality of the data, the number of support vectors discovered can be used to compute a (upper) bound on the anticipated error rate of the SVM classifier. Thus, even when the data's dimensionality is high, an SVM with a modest number of support vectors can have sound generalization[43-46].

VI. Experiments and Results

A. Data Collection

There are many free benchmarking English datasets utilized for documents classification, such as Reuters 21578, which contains 21,578 documents belonging to 17 classes; 20 Newsgroup, which contains around 20,000 documents distributed almost evenly into 20 classes; and RCV1 (Reuters Corpus Volume 1), which contains 806,791 documents classified into four main classes. Unfortunately, there is no free benchmarking dataset for Arabic documents

classification. For most Arabic document's classification research, authors collect their own datasets, mostly from online news sites.

We examine whether the use of document classification to evaluate what the best method for Arabic document summarization and to evaluate the impact of the summarization techniques on the accuracy of Arabic document classification. but What concerns us here is the use of documents classification to get the best way to summarize the documents, we have used an in-house dataset collected from several published papers for Arabic document classification and from scanning the well-known and reputable Arabic websites.

The collected corpus contains 5000 documents divided into ten categories of News, Economy, Health, History, Sport, Religion, Social, Nutriment, Technology and Law that vary in length and contain 500 documents in every category.

B. Experimental Configuration and Performance Measure

In this study, the benchmarking datasets mentioned earlier in the previous section need a set of preprocessing routines to be suitable for classification implementation. The Whole documents in the dataset were prepared by converting them to UTF 8 encoding and removing stop words. For stemming process, we used light stemming algorithm. In this study, the linear kernel for SVM classifier was applied because it has been clarified that the most classification problems are linearly separable [39]. The documents in the dataset are divided into two parts, 70% as training set and 30% as testing set. The training process use the training set and proposed classification algorithm to obtain a classification model that will be evaluated by means of the testing set.

C. Evaluation measure

We employ the standard measures to evaluate the performance of text classification before and after summarization process, i.e. precision, recall and F1-measure.

- Precision (P) is the number of sentences occurring in both system and ideal summaries divided by the number of sentences in the system summary.

$$Precision = \frac{TP}{TP + FP} \qquad (20)$$

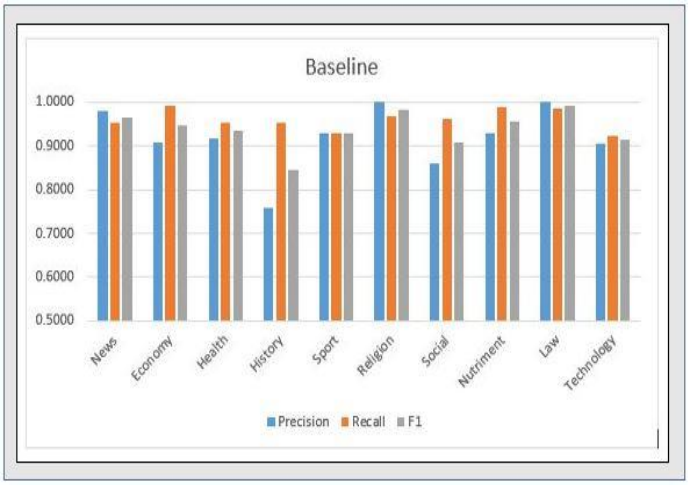
- Recall (R) is the number of sentences occurring in both system and ideal summaries divided by the number of sentences in the ideal summary.

$$Recall = \frac{TP}{TP + FN} \qquad (21)$$

Where:

TP is the number of documents that are correctly assigned to the category,

TN is the number of documents that are correctly assigned to the negative category,



FP is the number of documents a system incorrectly assigned to the category, Recall (R) is the proportion of predicted positive members among all actual positive class members in the data.

FN are the number of documents that belonged to the category but are not assigned to the category.,

- Micro -F1 (F1) The success measure, namely, micro-F1 score, a well-known F1 measure, is selected for this study, which is calculated as follows:

$$Micro - F1 = \frac{2 * Precision * Recall}{Precision + Recall} \qquad (22)$$

D. Experimental results and analysis

Six different experiments were conducted, one using original data and the rest using five different representations of the same dataset. The beginning experiment we used the original dataset without any summarization techniques and represented a baseline. The first representation used Continuous LexRank summarization technique as representatives of documents. The second used Centroid-Based summarization technique as representatives of documents. The third used Degree Centrality summarization technique as representatives of documents.

The fourth used LexRank summarization technique as representatives of documents. The last representation used Lakhas summarization technique. These representations were evaluated in terms of classification accuracy and execution time by using SVM classifier. The results of experiments that used the original dataset without any summarization techniques are illustrated in Table 1-6 and Figures 7-12, respectively. As to the rest of the results that shown classification accuracy after used summarization techniques, we will illustrate them successively through the tables and shapes following:

TABLE I. RESULTS OF TEXT CLASSIFICATION APPLIED TO: 1- (PRECISION, RECALL AND F-SCORE), AND 2- SUMMARIES CONTINUOUS LEXRANK

SVM Classifier	Baseline		
	Precision	Recall	F1
Economy	98.08%	95.21%	0.9662
Health	90.84%	99.19%	0.9483
History	91.77%	95.27%	0.9349
Law	75.94%	95.45%	0.8458
News	93.06%	93.01%	0.9303
Nutriment	100.00%	96.77%	0.9836
Religion	85.93%	96.15%	0.9075
Social	92.95%	98.80%	0.9579
Sport	100.00%	98.68%	0.9934
Technology	90.44%	92.25%	0.9134
Averages	0.9190	0.9608	0.9381
Macro F1	0.9381		
Micro F1	0.9607		

Fig. 7. Text Classification before Summarization (Baseline).

Figure 7 shows text classification (classifying the full documents) before summarization (Baseline), Our results show that the classification accuracy was 0.9607,

TABLE II. RESULTS OF TEXT CLASSIFICATION APPLIED TO: 1- (PRECISION, RECALL AND F-SCORE), AND 2- SUMMARIES CONTINUOUS LEXRANK

SVM Classifier	Continuous LexRank		
	Precision	Recall	F1
Economy	0.9744	0.9102	0.9412
Health	0.9008	0.9593	0.9291
History	0.9797	0.8580	0.9148
Law	0.7967	0.9416	0.8631
News	0.9437	0.9371	0.9404
Nutriment	1.0000	0.9806	0.9902
Religion	0.7987	0.9462	0.8662
Social	0.9795	0.8614	0.9167
Sport	1.0000	0.9669	0.9832
Technology	0.8671	0.8732	0.8701
Averages	0.9241	0.9235	0.9215
Macro F1	0.9215		
Micro F1	0.9213		

TABLE III.

ABLE TYPE STYLES

TABLE IV.

Table Head	Table Column Head		
	Table column subhead	*Subhead*	*Subhead*
copy	More table copy[a]		

Fig. 8. Summary Classification by Continuous LexRank method.

Figure 8 shows text Classification (classifying the summary documents) after summarization by used Continuous LexRank method, our results show that the classification accuracy was 0.9213,

TABLE III. RESULTS OF TEXT CLASSIFICATION APPLIED TO: 1- (PRECISION, RECALL AND F-SCORE), AND 2- SUMMARIES CENTROID BASED

SVM Classifier	Centroid Based		
	Precision	Recall	F1
Economy	0.9811	0.9341	0.9570
Health	0.96	0.9756	0.9677
History	0.9762	0.9704	0.9733
Law	0.96	0.9351	0.9474
News	0.9195	0.958	0.9384
Nutriment	1	1	1.0000
Religion	0.9407	0.9769	0.9585
Social	1	0.994	0.9970
Sport	1	0.9934	0.9967
Technology	0.9444	0.9577	0.9510
Averages	0.9682	0.9695	0.9687
Macro F1	0.9687		
Micro F1	0.9693		

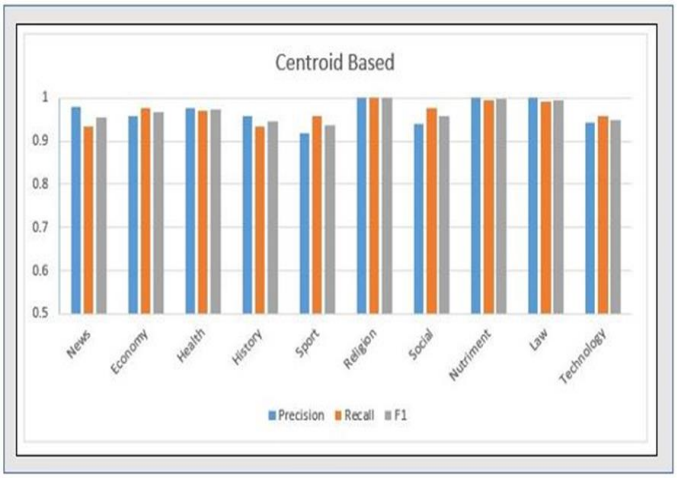
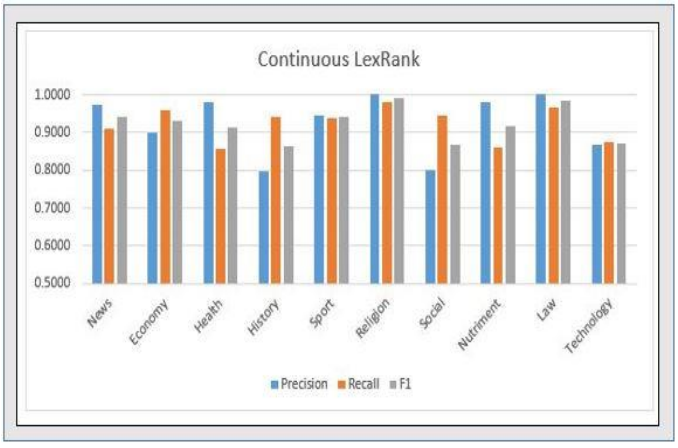


Fig. 9. Summary Classification by Centroid Based method



SVM Classifier	Lakhas		
	Precision	Recall	F1
Economy	0.9811	0.9341	0.9570
Health	0.9590	0.9512	0.9551
History	0.9618	0.8935	0.9264
Law	0.8503	0.9221	0.8847
News	0.9133	0.9580	0.9351
Nutriment	1.0000	0.9859	0.9929
Religion	0.9038	0.9097	0.9067
Social	0.9921	0.9615	0.9766
Sport	1.0000	0.9940	0.9970
Technology	0.9051	0.9470	0.9256
Averages	0.9467	0.9457	0.9457
Macro F1	0.9457		
Micro F1	0.9437		
News	0.7594	0.9221	0.8329
Nutriment	0.9306	0.9371	0.9338
Religion	1.0000	0.9742	0.9869
Social	0.8593	0.8923	0.8755
Sport	0.9295	0.8735	0.9006
Technology	1.0000	0.9669	0.9832
Averages	0.9044	0.8662	0.8849
Macro F1	0.9169		
Micro F1	0.916		

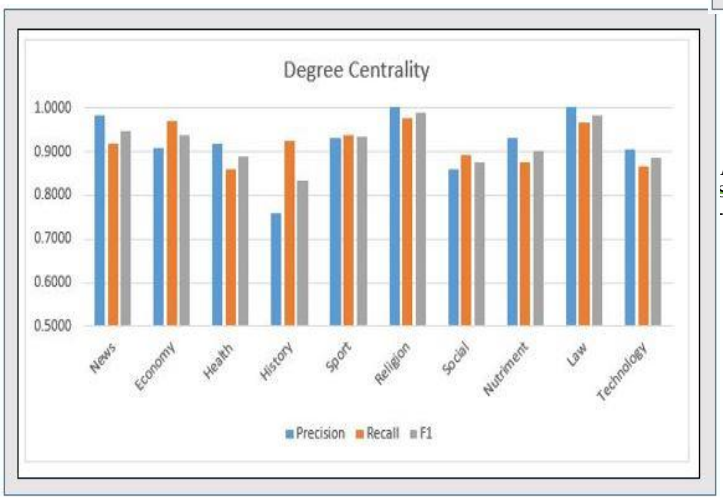


Fig. 10. Summary Classification by Degree Centrality method.

Figure 10 shows text classification (classifying the summary documents) after summarization by used degree centrality method, our results show that the classification accuracy was 0.916,

TABLE V. RESULTS OF TEXT CLASSIFICATION APPLIED TO: 1- (PRECISION, RECALL AND F-SCORE), AND 2- SUMMARIES LEXRANK

SVM Classifier	LexRank		
	Precision	Precision	Precision
Economy	0.9875	0.9875	0.9875
Health	0.9444	0.9444	0.9444
History	0.9367	0.9367	0.9367
Law	0.7956	0.7956	0.7956
News	0.9429	0.9429	0.9429
Nutriment	1.0000	1.0000	1.0000
Religion	0.8227	0.8227	0.8227
Social	0.9808	0.9808	0.9808
Sport	1.0000	1.0000	1.0000
Technology	0.9000	0.9000	0.9000
Averages	0.9311	0.9311	0.9311
Macro F1	0.9293		
Micro F1	0.9293		

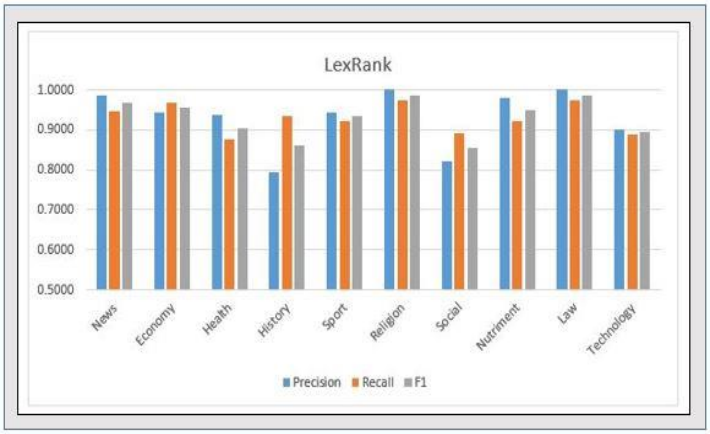


Fig. 11. Summary Classification by LexRank method.

ABLE VI. R
SULTS OF TEXT CLASSIFICATION APPLIED TO: 1- (PRECISION, RECALL AND
-SCORE), AND 2- SUMMARIES LAKHAS

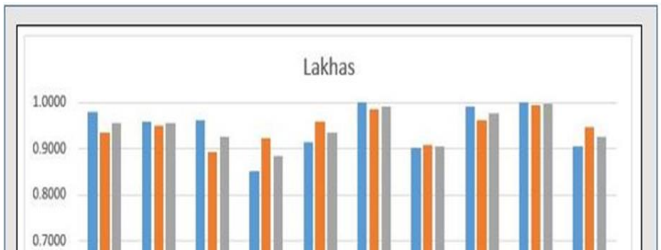


Fig. 12. Summary Classification by Lakhas method.

Figure 12 shows text Classification (classifying the summary documents) after summarization by used Lakhas method, our results show that the classification accuracy was 0.9437,

TABLE VII.
OMPARATIVE CLASSIFICATION ACCURACY AND TIME EXECUTE

SVM Classifier	Accuracy	Time Execution (Minute)
Baseline	0.9607	05:39
Continuous LexRank	0.9213	03:03
Degree Centrality	0.916	03:55
LexRank	0.9293	03:57
Lakhas	0.9447	04:06
Centroid Based	0.9693	03:42

Fig. 13. Comparative among summarization methods by Accuracy and Time.

Figure 13 shows Comparative among summarization methods different from through classification Accuracy and the time of execution and as it is shown also in Table 7. where the results showed that Centroid Based summarizer significantly outperformed the other summarization techniques from where Accuracy and provides significant improvement on the accuracy document classification.

REFERENCES

[1] Lloret, E. and M. Palomar, Text summarisation in progress: a literature review. Artificial Intelligence Review. 37(1): p. 1-41.' 2012.

[2] Mihalcea, R. and S. Hassan, Using the essence of texts to improve document classification.' 2005.

[3] Aggarwal, C.C. and C. Zhai, A survey of text classification algorithms, in Mining text data. 2012, Springer. p. 163-222.' 2012.

[4] Farid, D.M., et al., Hybrid decision tree and naïve Bayes classifiers for multi-class classification tasks. Expert Systems with Applications. 41(4): p. 1937-1946.' 2014.

[5] Ghiassi, M., et al., Automated text classification using a dynamic artificial neural network model. Expert Systems with Applications. 39(12): p. 10967-10976.' 2012.

[6] Lee, L.H., et al., High Relevance Keyword Extraction facility for Bayesian text classification on different domains of varying characteristic. Expert Systems with Applications. 39(1): p. 1147-1155.' 2012.

[7] Nehar, A., et al. An efficient stemming for arabic text classification. in Innovations in Information Technology (IIT), 2012 International Conference on. of Conference.: IEEE.' 2012.

[8] Ramanujam, N. and M. Kaliappan, An Automatic Multidocument Text Summarization Approach Based on Naïve Bayesian Classifier Using Timestamp Strategy. The Scientific World Journal. 2016.' 2016.

[9] Al-Saleh, A.B. and M.E.B. Menai, Automatic Arabic text summarization: a survey. Artificial Intelligence Review. 45(2): p. 203-234.' 2016.

[10] Haywood, J.A. and H.M. Nahmad, A new Arabic grammar of the written language. 1962: Lund, Humphries.' 1962.

[11] Shen, D., et al. Web-page classification through summarization. in Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval. of Conference.: ACM.' 2004.

[12] Berger, A.L. and V.O. Mittal. OCELOT: a system for summarizing Web pages. in Proceedings of the 23rd annual international ACM C SIGIR conference on Research and development in information retrieval. of Conference.: ACM.' 2000.

[13] Buyukkokten, O., H. Garcia-Molina, and A. Paepcke. Seeing the whole in parts: text summarization for web browsing on handheld devices. in Proceedings of the 10th international conference on World Wide Web. of Conference.: ACM.' 2001.

[14] Delort, J.-Y., B. Bouchon-Meunier, and M. Rifqi. Web Document Summarization by Context. in WWW (Posters). of Conference.' 2003.

[15] Shen, D., Q. Yang, and Z. Chen, Noise reduction through summarization for Web-page classification. Information processing & management. 43(6): p. 1735-1747.' 2007.

[16] Luhn, H.P., The automatic creation of literature abstracts. IBM Journal of research and development. 2(2): p. 159-165.' 1958.

[17] Ferreira, R., et al. Automatic Document Classification using Summarization Strategies. in Proceedings of the 2015 ACM Symposium on Document Engineering. of Conference.: ACM.' 2015.

[18] Al-Hindi, K. and E. Al-Thwaib, A comparative study of machine learning techniques in classifying full-text Arabic documents versus summarized documents. World of Computer Science and Information Technology Journal (WCSIT). 2(7): p. 126-129.' 2013.

[19] MARLENE, G., VERGHESE D., and e. al., TEXT CLASSIFICATION WITH SUMMARIES GENERATED USING LATENT SEMANTIC ANALYSIS., International Journal of Research Sciences and Advanced Engineering [IJRSAE] TM p. 264 - 271,.' 2014.

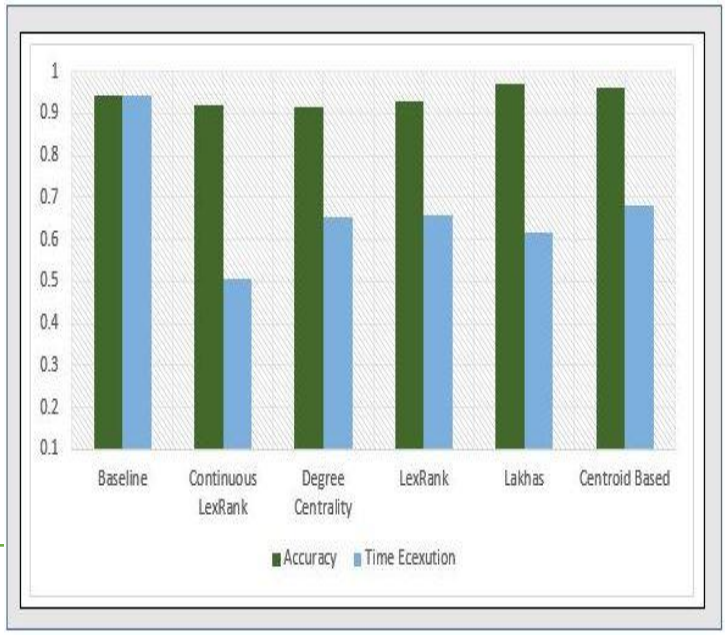
[20] Mamakis, G., et al., Document Classification in Summarization. Journal of Information and Computing Science. 7(1): p. 025-036.' 2012.

[21] Edmundson, H.P., New methods in automatic extracting. Journal of the ACM (JACM). 16(2): p. 264-285.' 1969.

[22] Schlesinger, J.D., D.P. O'leary, and J.M. Conroy. Arabic/English multi-document summarization with CLASSY—the past and the future. in International Conference on Intelligent Text Processing and Computational Linguistics. of Conference.: Springer.' 2008.

[23] Douzidia, F.S. and G. Lapalme, Lakhas, an Arabic summarization system. Proceedings of DUC2004.' 2004.

[24] Kogilavani, A. and P. Balasubramani, Clustering and feature specific sentence extraction based summarization of multiple documents. International Journal of Computer Science Information Technology. 2(4): p. 99-111.' 2010.



[25] Perumal, P. and R. Nedunchezhian, Performance Evaluation of Three Model-Based Documents Clustering Algorithms. European journal of Scientific Research. 52(4): p. 618-628.' 2011.

[26] Algaphari, G., F.M. Ba-Alwi, and A. Moharram, Text Summarization using Centrality Concept. International Journal of Computer Applications. 79(1).' 2013.

[27] Erkan, G. and D.R. Radev, Lexrank: Graph-based lexical centrality as salience in text summarization. Journal of Artificial Intelligence Research. 22: p. 457-479.' 2004.

[28] Radev, D.R., H. Jing, and M. Budzikowska. Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation, and user studies. in Proceedings of the 2000 NAACL-ANLP Workshop on Automatic summarization. of Conference.: Association for Computational Linguistics.' 2000.

[29] Radev, D.R., S. Blair-Goldensohn, and Z. Zhang, Experiments in single and multi-document summarization using MEAD. Ann Arbor. 1001: p. 48109.' 2001.

[30] Brin, S. and L. Page, The anatomy of a large-scale hypertextual web search engine. Computer networks and ISDN systems. 30(1): p. 107-117.' 1998.

[31] Aihua, L. and A. Lei, Research on the m-text automatic summarization optimization based on continuous lexrank. International Journal of Advancements in Computing Technology ,5.' 2013.

[32] Baharudin, B., L.H. Lee, and K. Khan, A Review of Machine Learning Algorithms for Text-Documents Classification. Journal of Advances in Information Technology. 1(1).' 2010.

[33] El Kourdi, M., A. Bensaid, and T.-e. Rachidi. Automatic Arabic document categorization based on the Naïve Bayes algorithm. in Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages. of Conference.: Association for Computational Linguistics.' 2004.

[34] Al-Saleem, S., Associative classification to categorize Arabic data sets. The International Journal Of ACM JORDAN. 1: p. 118-127.' 2010.

[35] Syiam, M.M., Z.T. Fayed, and M.B. Habib, An intelligent system for Arabic text categorization. International Journal of Intelligent Computing and Information Sciences. 6(1): p. 1-19.' 2006.

[36] Bawaneh, M.J., M.S. Alkoffash, and A. Al Rabea, Arabic Text Classification using K-NN and Naive Bayes. Journal of Computer Science. 4(7): p. 600-605.' 2008.

[37] Moh'd A Mesleh, A., Chi square feature extraction based SVMs Arabic language text categorization system. Journal of Computer Science. 3(6): p. 430-435.' 2007.

[38] Alaa, E., A COMPARATIVE STUDY ON ARABIC TEXT CLASSIFICATION. researchgate. net.' 2008.

[39] Joachims, T., Text categorization with support vector machines: Learning with many relevant features. 1998: Springer.' 1998.

[40] Burges, C.J., A tutorial on support vector machines for pattern recognition. Data mining and knowledge discovery. 2(2): p. 121-167.' 1998.

[41] Cortes, C. and V. Vapnik, Support-vector networks. Machine learning. 20(3): p. 273-297.' 1995.

[42] Hsu, C.-W., C.-C. Chang, and C.-J. Lin, A practical guide to support vector classification.' 2003.

[43] Duda, R.O., P.E. Hart, and D.G. Stork, Pattern classification. 2nd. Edition. New York.' 2001

[44] Han, J., J. Pei, and M. Kamber, Data mining: concepts and techniques. 2011: Elsevier.' 2011.

[45] Witten, I.H. and E. Frank, Data Mining: Practical machine learning tools and techniques. 2005: Morgan Kaufmann.' 2005

[46] Fan, R.-E., et al., LIBLINEAR: A library for large linear classification. Journal of machine learning research. 9(Aug): p. 1871-1874.' 2008.