ARABIC TEXT CLASSIFICATION USING SMO, NAÏVE BAYESIAN, J48 ALGORITHMS

Majed Ismail Hussien1, Fekry Olayah2, Minwer AL-dwan3 & Ahlam Shamsan4
1Al- Isra University (drmajed1@hotmail.com)
2Al- Isra University, (fekrydrm@yahoo.com)
3AL-Ahliyah Amman University (Jusafe2001@yahoo.com)
4Amman Arab University, (ahlam_shamsan@yahoo.com)

ABSTRACT
Several algorithms have been implemented to solve the problem of text categorization. Most of the work in this area was performing for English text, while few researches have been performing for the Arabic text. However, the nature of Arabic text is different than English text; preprocessing of Arabic text and more challenging. This paper implemented the Sequential Minimal Optimization (SMO), Naïve Bayesian (NB) and J48 (C4.5) Algorithms using weka program, comparing between the algorithms in accuracy and time to get the result. A huge number of features or keywords in the documents lead to a poor performance in terms of both accuracy and time. Therefore preprocessing is very important step before the categorization documents to get knowledge from massive data and reduce the processing operations. The preprocessing includes two approaches: the first elimination stop word and the second normalization approach. The results show that the (SMO) classifier achieves the highest accuracy and the lowest error rate, followed by J48 (C4.5), then the (NB) classifier. But the second part of the results (time) shows that the time needed to get the results, the SMO model is the faster one, followed by NB model, and then J48 classifier which takes a highest amount of time.

Keywords: Accuracy, error rate, stopwards, document preprocessing, categorizations algorithms.

1. INTRODUCTION
With the fast growth of online information, text classification has become one of the key techniques for handling and organizing text data. Text classification techniques are used to search information on the WWW, and to find interesting information in it. Text classification (or categorization) is the process of structuring a set of documents according to a group structure to get a set of one or more categories [1], the organization of text in categories allows the user to limit the target of a search submitted to Information Retrieval Systems (IRS), to explore the collection data and to find relevant information to user needs with poor knowledge about the keywords of a them.

There are several different methods for text classification, including, Bayesian classification, distance-based algorithms, decision tree-based methods [2]. Developing text classification systems for Arabic documents is a challenging task due to the complex and rich nature of the Arabic language. The Arabic language consists of 28 letters, written from right to left. It has very complex morphology, and the majority of words have a tri-letter root. The rest have either a quad-letter root, penta-letter root or hexa-letter root [3].

Text classification is a well-known task in data mining that aims to predict the class of an unseen instance as accurately as possible. There are many types of techniques that are used to classification of text, each one has specific characteristic. The different between them are based on the degree of accuracy.

According to [4] used K-NN and Naïve Bayes to classification Arabic text, they used different k values starting from 1 and up to 20 in order to find the best results for KNN. Effectiveness started to decline at k>15.

The k-fold cross-validation method is used to test the accuracy. The program makes comparison between the two algorithms, and the results that the k-nearest neighbors is better than Naïve Bayesian classifier.

In additional [5] compared between three classifier for Arabic text, the naive Bayes, k-nearest-neighbor (knn), and distance-based classifier, and compared the accuracy using recall, precision, error rate and fallout. The result of the naive Bayes classifier outperforms the other two.

Another study [3] presented the results of classifying Arabic text documents using the N-gram frequency statistics technique employing a dissimilarity measure called the “Manhattan distance”, and Dice’s measure of similarity. The Dice measure was used for comparison purposes. Results showed that N-gram text classification using the Dice measure outperforms classification using the Manhattan measure.

According to [6] proposed a novel batch-updated approach to enhance the performance of Centroid Classifier to be a simple and efficient method. Because often suffers from misfit incurred by its assumption. The main idea is to take advantage of training errors to sequentially update the classification model by group. The technique is simple to implement and flexible. The results indicate that the technique can significantly improve the
performance of Centroid Classifier. In this paper will be use Naïve Bayesian, SMO and J48 classifier to categorize the Arabic text and get accuracy results from each classifier and how long time will take for each classifier. In the next sections will be describe the text classification process, the running experiment then the conclusion and appendix of application figures.

2. PROPOSED TEXT CLASSIFICATION PROCESS:
This section describe the main phases of the text classification process start with the data set which consist of 2363 documents, divided to six categories based on its contents. The next step is the documents preprocessing is very important step before the categorization documents to get knowledge from massive data and reduce the processing operations. The preprocessing includes two approaches: the first elimination stop word and the second normalization approach. After that document tokenizing which means Analyze text into a sequence of discrete tokens (words) then stored it in the document without repeat. The next step is classify the text by SMO, NB, J48 algorithms using two methods to gets the accuracy and time needed to built the models; the first method is percentage split and the second is 10 fold cross validation method. Then the final step is evaluating the efficiency of accuracy and time needed results, as show in figure (1).

2.1 Data Set:
The data set consist of 2363 documents of different lengths. Each document was manually labeled based on its contents and the domain that it was found within, each document is stored in a separate file; these documents categorized to six categories (Sport, Economic, Medicine, Politic, Religion and Science) as show in table (1).

<table>
<thead>
<tr>
<th>Category name</th>
<th># of documents</th>
</tr>
</thead>
<tbody>
<tr>
<td>رياض (sport)</td>
<td>466</td>
</tr>
<tr>
<td>اقتصاد (economic)</td>
<td>93</td>
</tr>
<tr>
<td>طبية (medicine)</td>
<td>42</td>
</tr>
<tr>
<td>سياسة (politic)</td>
<td>1040</td>
</tr>
<tr>
<td>دين (religion)</td>
<td>324</td>
</tr>
<tr>
<td>وعلم (science)</td>
<td>391</td>
</tr>
</tbody>
</table>

2.2 Preprocessing the Documents
A huge number of features or keywords in the documents lead to a poor performance in terms of both accuracy and time [5]. Therefore preprocessing is very important step before the categorization documents to get knowledge from massive data and reduce the processing operations. The preprocessing includes two
approaches: the first elimination stop word and the second normalization approach.

- **Elimination of Stop words**

Elimination of stop words concern to remove of non meaningful words which don’t indicate the semantic content of the document, these stop words have two different impacts on the information retrieval process. First, they have a very high frequency and tend to reduce the impact of frequency differences among less common words, second affecting the weighting process. The remove of the stop words also changes the document length; reduce the memory and recall process [7]. If a word which occurs in 80% of the documents in the collection is useless for purposes of retrieval, such as stop words.

Some words appear in the sentences and don’t have any meaning or indications about the content such as (so, with, for, confirmation, or appearing frequently in the document like pronouns such as (he, she, they, to) or prepositions such as (from, in) or demonstratives such as (this, these, who, which, who, who, who, who), there are many types of stop words.

Also the numbers and symbols like (@, #, %, *) and some words that indicates a sequence of the sentences such as (firstly, secondly, третьий), Some Arabic documents may contain foreign words, especially science documents, these words are not important for us so these words has to be ignored by the system [7]. All this types of words may have bad effect to the information retrieval system to candidate the keywords.

In this research used tool in .net studio in C# language to remove the stop words from the text as show in figure (2).

![Figure 2 preprocessing the Documents](image)

After removing all the above types of stop words we have pure sentences with meaningful words and from these sentences the system will extract candidate keywords.

- **Documents Normalization**

Normalization is important stage that will reduce many terms that have the same meaning but they written in different forms and this is common issue in Arabic language, for example of single word written in many forms (أحمد حسن محمد).

Two methods available for documents normalization expansion or reduction, in this research will be used the token reduction approach for own documents. The benefits of this step is minimizing storage requirements by eliminating redundant terms, as well as increasing matching probability for document [8].

2.3 The implementation

After pre-processing will transform the documents into the vector model which represents documents as vectors in the space, each vector can be represented by the weights of terms in a document with respect to the dimension of the space, the number of dimensions equals the number of terms or keywords used, then we have start the classify the data by algorithms which we chosen (SMO, Bayesian, J48) then the classifier splits the
data set into two subsets:
- The training subset: where k documents are used as examples for the classifier, and must contain sufficient number of positive examples for all the categories involved.
- The testing subset: used to test the classifier effectiveness [9].

After classify data we measure the accuracy by recall and precision measure, recall is the number of items of category identified divide by the number of category members in test set. Precision is the number of items of category identified.

3. **RUNNING EXPERIMENTS**:

The dataset was tested using two methods for measuring accuracy.

Percentage split method (holdout), where 60% of the data used as training and the remaining 40% used as testing.

- K-fold cross validation methods, the data was divided into 10 folds, some fold is used as testing and the remaining folds are used as training.

### 3.1 Measure of accuracy and error rate before stopwords eliminate

The results of accuracy and error rate before stopwords eliminate as follows:

<table>
<thead>
<tr>
<th></th>
<th>Accuracy</th>
<th>Error rate</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SMO</td>
<td>NB</td>
</tr>
<tr>
<td>Percentage Split 60%</td>
<td>94.8%</td>
<td>85.07%</td>
</tr>
<tr>
<td>10 Folds CV</td>
<td>96.4%</td>
<td>83.87%</td>
</tr>
</tbody>
</table>

The SMO (SVM) classifier achieves the highest accuracy (94.8%), (96.61%) and the lowest error rate (5.2%), (3.6%) using percentage split.

On the other hand the results in J48 classifier less accurate than that in SMO classifier.

But the NB classifier achieves the lowest accuracy (85.07%), (83.87%) and the highest error rate (14.92%), (16.12%) comparing with other two classifiers.
Sorting the classifiers according to their performance from the higher to the lower:
Naïve Bayes Classifier

3.2 Measure of time before stopwords eliminate
Here measure the amount of time that taken for building the models that’s used for testing the accuracy of the classifiers.

Table (3) Time needed to build the models in seconds

<table>
<thead>
<tr>
<th></th>
<th>SMO</th>
<th>NB</th>
<th>J48</th>
</tr>
</thead>
<tbody>
<tr>
<td>Percentage Split</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>60%</td>
<td>7.94</td>
<td>15.25</td>
<td>261.81</td>
</tr>
<tr>
<td>10 folds CV</td>
<td>7.52</td>
<td>20.98</td>
<td>263.61</td>
</tr>
</tbody>
</table>

From the table above the SMO classifier is the faster one to build the needed models that is used for testing the
accuracy of the classifier, while NB classifier the second one, but the J48 requires a lot of time to build the needed model.

Sorting the classifiers according to the time needed to build the models from the smallest time needed:

SMO (SVM) Classifier, Naïve Bayes Classifier, J48 Classifier.

3.3 Measure of accuracy and error rate after stop word elimination

The results of accuracy and error rate after stopwords eliminate as follows:

Table (4) Accuracy and Error rate after stop words eliminate

<table>
<thead>
<tr>
<th></th>
<th>Accuracy</th>
<th>Error rate</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SMO</td>
<td>NB</td>
</tr>
<tr>
<td>Percentage</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Split 60%</td>
<td>96.08%</td>
<td>85.60%</td>
</tr>
<tr>
<td></td>
<td>3.42%</td>
<td>14.4%</td>
</tr>
<tr>
<td>10 Folds</td>
<td>96.64%</td>
<td>83.79%</td>
</tr>
<tr>
<td>CV</td>
<td>3.35%</td>
<td>16.21%</td>
</tr>
</tbody>
</table>

According to the table above the results after eliminate the stopwords are more accurate than results before eliminate it.

3.4 Measure of time After stop words elimination

Also the time needed to built the models after eliminate the stopwords is less than before eliminate it.
The amount of time to test the accuracy after eliminate the stopwords are less than the results before eliminate it.

### 3.5 Measure of Precision and Recall
Figure (6) shows measure of recall before and after stopwords eliminate but the result after eliminate the stopwords is better than before.

Figure (6) Recall measure before and after stopwords eliminate

Figure (7) shows measure of precision before and after stopwords eliminate also the result after eliminate the stopwords is better than before.

Figure (7) Precision before and after Stopwords Eliminations

### 4. CONCLUSION
Several algorithms have been implemented to solve the problem of text categorization. Most of the work in this area was performing for the English text, while very few researches have been performing for the Arabic text. The study compares between three classification techniques for Arabic text, the results show that there are differences between the classifiers from three aspects. (The accuracy, error rate, and time taken to build the classification model). The results show that the Sequential Minimal Optimization (SMO) classifier achieves the highest accuracy and the lowest error rate, followed by the J48 (C4.5) classifier, then by the Naive Bayes (NB) classifier. The second part of the results shows that the time needed to build the SMO model is the faster one, followed by NB model, then J48 classifier which takes a highest amount of time.

5. REFERENCES
6. APPENDIX

1. SMO application figures
2. **NB application figures**
3. **J48 application figures**

<table>
<thead>
<tr>
<th>True Positive</th>
<th>True Negative</th>
<th>False Positive</th>
<th>False Negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>90</td>
<td>10</td>
<td>0</td>
</tr>
</tbody>
</table>

**Summary**

- **Correctly Classified Instances**: 2100
- **Percent Correct**: 95.06%
- **Confusion Matrix**
  - **Class**:
    - Economy: 100
    - Religion: 90
    - Religion: 10
    - Religion: 0

**Detailed Accuracy By Class**

<table>
<thead>
<tr>
<th>Class</th>
<th>TP Rate</th>
<th>FP Rate</th>
<th>Precision</th>
<th>Recall</th>
<th>F-Measure</th>
<th>ROC Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>Economy</td>
<td>1.00</td>
<td>0.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>Religion</td>
<td>0.90</td>
<td>0.10</td>
<td>0.90</td>
<td>0.90</td>
<td>0.90</td>
<td>0.90</td>
</tr>
</tbody>
</table>

**Confusion Matrix**

<table>
<thead>
<tr>
<th></th>
<th>Economy</th>
<th>Religion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Economy</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>Religion</td>
<td>0</td>
<td>90</td>
</tr>
</tbody>
</table>

**Test Set Statistics**

- **Number of Instances**: 2000
- **Incorrect Classification**: 100
- **Confusion Matrix**
  - **Class**:
    - Economy: 100
    - Religion: 90
    - Religion: 10
    - Religion: 0

**Detailed Accuracy By Class**

<table>
<thead>
<tr>
<th>Class</th>
<th>TP Rate</th>
<th>FP Rate</th>
<th>Precision</th>
<th>Recall</th>
<th>F-Measure</th>
<th>ROC Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>Economy</td>
<td>0.95</td>
<td>0.05</td>
<td>0.95</td>
<td>0.95</td>
<td>0.95</td>
<td>0.95</td>
</tr>
<tr>
<td>Religion</td>
<td>0.90</td>
<td>0.10</td>
<td>0.90</td>
<td>0.90</td>
<td>0.90</td>
<td>0.90</td>
</tr>
</tbody>
</table>

**Confusion Matrix**

<table>
<thead>
<tr>
<th></th>
<th>Economy</th>
<th>Religion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Economy</td>
<td>95</td>
<td>5</td>
</tr>
<tr>
<td>Religion</td>
<td>5</td>
<td>90</td>
</tr>
</tbody>
</table>

**Test Set Statistics**

- **Number of Instances**: 2000
- **Incorrect Classification**: 100
- **Confusion Matrix**
  - **Class**:
    - Economy: 100
    - Religion: 90
    - Religion: 10
    - Religion: 0