A Comparative Study of Automatic Text Categorization Methods Using Arabic Text

Ismail Hmeidi, Majd Al-shalabi and Mahmoud Al-Ayyoub
Jordan University of Science and Technology
Irbid, Jordan
hmeidi@just.edu.jo, majd_cs@yahoo.com, maalshbool@just.edu.jo

ABSTRACT
There are several researches and procedures for classifying Arabic-language texts that are based mostly on different environments. This lack of dependence on a unified standard (such as a unified dataset) makes it hard to determine the most accurate technique for classification. In this paper, we study and analyze the classification algorithms based on a unified environment and a different dataset with the included challenges faced by these algorithms to demonstrate their effectiveness and accuracy with a large dataset.

Keywords: Arabic Text Classification; Naive Bayes; Decision Tree; KNN

1. INTRODUCTION
A tremendous amount of pages and topics in the online world are becoming accessible to everyone. Users can easily write threads and upload files onto web pages giving people a great opportunity to share massive amounts of data. However, such advances gave rise to new challenges such as the ability to retrieve the required piece of information efficiently and effectively. Retrieving what the user wants or even the closest topics to the user’s request, is at the core of the information retrieval discipline. What makes this problem complex and challenging is the large number of existing topics with overlapping terminology.

In the world of computer and internet, there must be solutions to these problems; otherwise the process of searching and retrieving information on the internet is useless and may take a long time to reach the user request. Here comes the importance of the classification text in order to facilitate the retrieval of the required information. There are many areas for subjects such as medicine, sports, health, law, etc. Narrowing down the search space by focusing on the domain in which the user is interested is likely to improve the information retrieval process.

The text classification is the automated technique used to classify the text in predefined category which is more related to the text. Part of the importance of text classification comes from its wide application. In addition to the traditional uses in information retrieval, other applications of text classification include spam filtering [5], sentiment analysis [1, 8, 3, 2, 12], determining author’s characteristics such as identity [26, 36, 14], gender [16, 13], dialect [42, 43], native language [39], political orientation [30, 4], etc.

Most research has focused on classifying texts written in the English language. Other languages such as Arabic received less interest due the nature of these languages and the difficulty of their structures. The difficult nature in the Arabic language makes it more complex and difficult to deal with them because of the many rules and anomalous characteristics, but it has become necessary to deal with this language because of the widespread over the Internet. To facilitate the search and retrieval in the Arabic language there are many algorithms working on the text classification that helps to retrieve data related to research in a short time and more accurate.

In this paper, we have studied many classification algorithms of Arabic language texts, there are many algorithms used for classification, but which of them is better? So, we chose some of the text and classification algorithms and we have applied it to the dataset written in Arabic language, each of these algorithms has the characteristics and standards, such as precision, Recall, F-measure and accuracy. Our problem in this paper is to find which the algorithm is the best depends on the result which collected from WEKA software and make comparisons between classifier algorithms. In these days Arabic texts spread largely due to the spread of the Internet and the ease of access and upload Arabic files dramatically, this led to the difficulty of research and an increase in the time to reach the required results. This is what makes researchers and programmers are looking for solutions to these developments to facilitate the search and retrieval operations. The huge increase in the number of Arabic texts on the Internet has increased the complexity, so the process of classifying texts working to improve the process of retrieving data, the researchers and programmers developed algorithms for the classification of Arabic-language texts, and there are many of them, and each of these algorithms have characteristics that distinguish them from other, such as the accuracy and precision and recall differ these characteristics depending on the nature of the data.

In this paper, we applied some algorithms in a different of Arabic dataset and make comparison between them to help to make the decision of what the algorithm that we will use and when can be used, depend on the results we have obtained from the comparison. Several algorithms, concerned the classification of texts, but differ in terms of accuracy in this paper that are interested in the process of classifying texts we have made a comparison between the algorithms for text classification in Arabic.

The rest of the paper is organized as follows. The following section gives a general overview of the current literature on text categorization with a focus on the Arabic language. Our work is discussed in Section 3. Finally, concluding remarks along with a discussion of future work is discussed in Section 4.

2. RELATED WORKS
Many researches are proposed and presented for the problem of the Arabic text classification. In this section we mention the main algorithms of these studies such as: Decision tree [7], KNN [27, 10, 41, 11], NB [40, 21, 15], N-Gram frequency [29, 28], Rocchio
There are several works and studies on text categorization of Arabic text and every work considers some points and leaves others depending on the type of study. In [19], the authors consider classification of Arabic text that is very robust and reliable without morphological analysis. In [20] the authors conduct a comparative study using N-Gram and using two measures, Manhattan measure and Dice’s measure. They compare them together and the result was that the N-Gram with Dice’s measure is better than using Manhattan measure. In [35] considers both labeled and unlabeled documents using expectation-maximization (EM). They proposed an algorithm based on EM with the Naive Bayes (NB) classifier to learn from the documents, both labeled and non-labeled.

Dwairi [18] used the distance-based, KNN, and NB classifiers to classify a set of Arabic documents collected from online magazines and news papers. The corpus contained 1000 documents that vary in length and writing styles with 10 predefined categories of equal size, 100 document per category. The holdout method was used to evaluate the classifier by dividing the corpus into two divisions each containing 50% of the document randomly chosen. The first part is used for training the classifiers and the remaining part is used for testing. For both training and testing, the documents were normalized by removing the punctuation mark, formatting tags, prepositions, pronouns, conjunctions, and auxiliary verbs from the words. A root-based stemmer adopted from [9] was used for stemming the normalized documents to reduce the number of words in the documents by extracting the roots of the words. The distance-based classifier was used to classify the test documents by building a feature vector for each category by adding the words, without redundancy, from the training documents that belong to that category. In other words, the feature vector for category Ci will contain the union of the words from the documents that belong to category Ci. At the end of the training phase, m feature vectors have been created of the words from the documents that belong to category Ci. Depending on the training set, m is the number of categories. To classify a new test document X, the classifier creates a feature vector for that document and calculates the similarity between it and the feature vector of each category. The dice similarity measure [23] was used to calculate the similarity between the feature vector of the test document and the feature vector of the categories. The test documents is assigned to the category with the highest similarity. The performance was evaluated in terms of precision, recall, fall-out, and error rate. Four different values were used for the KNN classifier which are 10, 20, 50, and 100. Results showed that NB classifier gave the highest accuracy, followed by KNN when k=50 and the distance-based classifier was the worst one with the minimum accuracy.

Hmeidi, Hawashin and El-Qawasmeh [25] studied the performance of the SVM and the KNN classifiers to classify a set Arabic article. The corpus was constructed from well known Jordanian newspapers called Alrai and Addustow. It contains two categories with 2260 documents for training the classifiers and 29 documents for the test. TF-IDF weighting approach was used to evaluate the importance of the words in the document by giving each word a weight. In order to reduce the number of words in the documents, the Chi square feature selection method [40, 25, 21] was used to select the words that best represent the documents. For the KNN classifier, the cosine similarity measure was used to calculate the distance between the training documents and test documents. The performance was evaluated in terms of precision, recall, and F1 measure. Various numbers of words: 50, 100, 150, 200, 250, 300, 350, 400, and 450 were selected to represent the documents of the corpus. Results showed that the SVM classifier gave better accuracy when the number of selected word is small, but the performance of the KNN classifier outperforms the SVM classifier when the numbers of selected words increase. Both classifiers reach the 100% accuracy when the number of selected words equal 450, this is due to the lack of the sufficient number of training and testing documents, where 98% of the documents were used for training the classifiers and only 2% of the documents were used for testing. Furthermore, the authors used full word classification which means that they did not use any preprocessing technique such as stemming and normalization to reduce the number of words in the documents. Using the full word article in documents classification leads to long classification time and sometimes decreases the classifier accuracy especially when using the KNN classifier.

Recently, several interesting works appeared addressing different aspects of the Arabic text classification problem. For example, the widespread of smart phones and online social networks gave rise to new styles of writing in which the text is short, cryptic, with little regard to spelling and grammatical rules and heavy use of slang, emoticons and symbols, etc. Addressing classification for texts with such characteristics pose different challenges than traditional text classification [22]. Another example, is the heavy use of multiple tags to describe each document. This gives rise to the multilabel text classification problem which is challenging on its own [6]. Finally, the focus on character-based features can give interesting ways of performing classification such as using compression tools [31]. A recent study [38] experimented with this idea for Arabic text.

3. METHODOLOGY

Most works have focused on classifying texts written in the English Language more than the Arabic language because of the Arabic nature and the difficulty of its structures. The difficult nature of the Arabic language makes it more complex and difficult to deal with because of the many rules and anomalous characteristics. However, it has become necessary to deal with this language because of its widespread usage online.

To facilitate the search and retrieval in the Arabic language there are many algorithms working on the text classification that helps to retrieve data related to research in a short time and high accuracy. In this work we study many classification algorithms of Arabic language texts.

In this paper, we choose some of the text and classification algorithms and apply them to the dataset written in Arabic language. Each of these algorithms have certain characteristics and standards, such as relative precision, recall, f-measure and accuracy. Our problem with this paper is to find when the algorithm is the best among the others depending on the results we collected from WEKA software.

3.1 Dataset

We use in this paper a dataset that is divided into five parts. Each part has nine categories: Art, Economy, Health, Law, Literature, Politics, Religion, Sport and Technology.

- Part 1: The original dataset without changes.
- Part 2: Dataset by removing stop words, punctuations and diacritics.
- Part 3: Dataset with applying the light 10 Stemmer.
- Part 4: Dataset with applying Chen stemmer.
Figure 1. Results when using files from Part 1.

Figure 2. Results when using files from Part 2.
Figure 3. Results when using files from Part 3.

(a) 600 files

(b) 1200 files

Figure 4. Results when using files from Part 4.

(a) 600 files

(b) 1200 files
Figure 5. Results when using files from Part 5.

Figure 6. Accuracy when using files from Part 1.
Figure 7. Accuracy when using files from Part 2.

(a) 600 files

(b) 1200 files

Figure 8. Accuracy when using files from Part 3.

(a) 600 files

(b) 1200 files
Figure 9. Accuracy when using files from Part 4.

Figure 10. Accuracy when using files from Part 5.
• Part 5: Dataset with applying Khuja algorithm for extracting the roots.

Classification algorithms were applied to each part, with applied features reduction. We experiment the algorithms using 600 and 1200 files for each category and then record the results for each experiment.

3.2 Experimental Results

We use in this paper the WEKA program to apply classification algorithms and compare their results in terms of True Positive (TP) Rate, False Positive (FP) Rate, Recall, Precision, F-Measure, ROC Area, Accuracy, and Time. The classification algorithms we consider are: Bayes network, KNN, decision tree, Kstar, Naive Bayes, Naive Bayes Multinomial and Random Forest.

3.2.1 Comparison among different algorithms

After classifying files using different the previously mentioned algorithms, and after recording the results in each time, we compare among the algorithms in terms we discussed before (TP, FP, etc.). Figure 1(a) describes the comparison among algorithms when we applied it using 600 files from Part 1 (original files). We see from this Figure that when we doubled the number of files that the best TP rate was obtained when we apply the Bayes Net algorithm, while the NB Multinomial algorithm comes next, and the last was KNN algorithm, either for other attributes we can see the Bayes Network algorithm is the best in the case of 600 files and original data.

Figure 1(b) describes the comparison among algorithms when we applied it using 1200 files from Part 1 (original files). We see from this Figure that the best TP rate was obtained when we apply the Random Forest algorithm, while the Kstar algorithm comes next, as for the precision, recall, and F-Measure we can see from the result the Random Forest algorithm is the best choice in the case of 1200 files.

Figure 2(a) describes the comparison among algorithms when we applied it using 600 files from Part 2 (with removing stop words, punctuations and diacritics). We see from this Figure that the best TP rate was obtained when we apply the NB Multinomial algorithm, while the Bayes Net algorithm comes next, when we duplicate the files we can see the change of results in this case NB Multinomial is the best algorithm when applied V2 with 600 files.

Figure 2(b) describes the comparison among algorithms when we applied it Using 1200 files from Part 2 (with removing stop words, punctuations and diacritics). We see from this Figure that the best TP rate was obtained when we apply the NB Multinomial algorithm, while the Bayes Net algorithm comes next, then Kstar algorithms, while the Random Forest come next then NB Multinomial, Bayes Net, and KNN algorithms come next, then Decision tree and the last one is Naive Bayes and the best FP rate was obtained when we apply the Kstar algorithms, while the Random Forest come next then NB Multinomial, Bayes Net, and KNN algorithms, then Decision tree and the last one is Naive Bayes, as for the precision was obtained when we apply the Kstar algorithm then Random Forest come next, as for the Recall, and F-Measure Kstar algorithm and Random Forest comes first and have best result at the case of 1200 files with Part 2.

Figure 3(a) describes the comparison among algorithms when we applied it using 600 files from Part 3 (with applying the light 10 stemmer). We see from this Figure that the best TP rate was obtained when we apply the NB Multinomial algorithm, while the Bayes Net algorithm comes next, as for the Precision, Recall, and F-Measure the NB Multinomial is good in this case.

Figure 3(b) describes the comparison among algorithms when we applied it using 1200 files from Part 3 (with applying the light 10 stemmer). We see from this Figure that the best TP rate was obtained when we apply the Random Forest algorithm, while the Kstar algorithm comes next, then Bayes Network algorithm then NB Multinomial and Decision Tree, as for the Naive Bayes come last one. As for the Precision the Random Forest algorithm has best average and the same about Recall and F-Measure.

Figure 4(a) describes the comparison among algorithms when we applied it using 600 files from Part 4 (with applying Chen stemmer). We see from this Figure that the best TP rate was obtained when we apply the NB Multinomial algorithm, while the Bayes Net algorithm comes next, as for the FP rate NB Multinomial algorithm and Bayes Network algorithm have small value that means the NB Multinomial have good accuracy in this case, we can see from the Figure the Precision value is high with the Decision Tree and we can see the KNN algorithm have small value, as for the Recall and F-Measure the NB Multinomial have high value and we can see that the KNN algorithm have small value.

Figure 4(b) describes the comparison among algorithms when we applied it using 1200 files from Part 4 (with applying Chen stemmer). We see from this Figure that the best TP rate was obtained when we apply the Random Forest algorithm, while the Kstar algorithm comes next, from the result, we can talk the Random Forest algorithm is the best in this case 1200 file v4.

Figure 5(a) describes the comparison among algorithms when we applied it using 600 files from Part 5 (with applying Khuja algorithm for extracting the roots). We see from this Figure that the best TP rate was obtained when we apply the NB Multinomial algorithm, while the Bayes Net algorithm comes next, as for the other attribute we can see that NB Multinomial have good average in this case.

Figure 5(b) describes the comparison among algorithms when we applied it using 1200 files from Part 5 (with applying Khuja algorithm for extracting the roots). We see from this Figure that the best TP rate was obtained when we apply the Random Forest algorithm, while the Kstar algorithm comes next, and as for the other attributes the Random Forest is a good algorithm for this case when applied on 1200 files and v5.

3.2.2 Accuracy Comparison

Figure 6(a) describes the comparison among algorithms when we applied them using 600 files from Part 1 (original files). Based on the Figure, we can clearly see that the highest accuracy is 96.625% and the lowest is 80.39%, the highest accuracy belongs to the Bayes Net, KNN at the bottom of the chart with percentage 44.01%.

Figure 6(b) describes the comparison among algorithms when we applied them using 1200 files from Part 1 (original files). Based on the Figure, we can clearly see that the highest accuracy is 99.13%, which belongs to Random Forest algorithm, and the lowest is 95.64% which belongs to NB algorithm.

Figure 7(a) describes the comparison among algorithms when we applied them using 600 files from Part 2 (with removing stop words, punctuations and diacritics). Based on the Figure, we can clearly see that the highest accuracy is 96.73% and the lowest is 79.09%, the highest accuracy belongs to the NB Multinomial, KNN at the bottom of the chart.

Figure 7(b) describes the comparison among algorithms when we applied them using 1200 files from Part 2 (with removing stop words, punctuations and diacritics). Based on the Figure, we can clearly see that the highest accuracy is 98.83% and the lowest is...
94.69%, the highest accuracy belongs to the Kstar, NB at the bottom of the chart.

Figure 8(a) describes the comparison among algorithms when we applied them using 600 files from Part 3 (with applying the light 10 stemmer). Based on the Figure, we can clearly see that the highest accuracy is 96.73%, which belongs to the NB Multinomial algorithm, and the lowest is 83.88% which belongs to KNN algorithm.

Figure 8(b) describes the comparison among algorithms when we applied them using 1200 files from Part 3 (with applying the light 10 stemmer). Based on the Figure, we can clearly see that the highest accuracy is 99.51%, which belongs to the Random Forest algorithm, and the lowest is 95.83% which belongs to NB algorithm.

Figure 9(a) describes the comparison among algorithms when we applied them using 600 files from Part 4 (with applying Chen stemmer). Based on the Figure, we can clearly see that the highest accuracy is 96.02%, which belongs to the NB Multinomial algorithm, and the lowest is 84.31% which belongs to KNN algorithm.

Figure 9(b) describes the comparison among algorithms when we applied them using 1200 files from Part 4 (with applying Chen stemmer). Based on the Figure, we can clearly see that the highest accuracy is 99.51%, which belongs to the Random Forest algorithm, and the lowest is 95.13% which belongs to NB algorithm.

Figure 10(a) describes the comparison among algorithms when we applied them using 600 files from Part 5 (with applying Khuja algorithm for extracting the roots). Based on the Figure, we can clearly see that the highest accuracy is 95.32% and the lowest is 83.61%, the highest accuracy belongs to the NB Multinomial, Decision Tree at the bottom of the chart with percentage 83.61%.

Figure 10(b) describes the comparison among algorithms when we applied them using 1200 files from Part 5 (with applying Khuja algorithm for extracting the roots). Based on the Figure, we can clearly see that the highest accuracy is 99.51% and the lowest is 94.25%, the highest accuracy belongs to the Random Forest, NB at the bottom of the chart with percentage 94.25%.

4. CONCLUSION AND FUTURE WORK

In the text classification there are some algorithms concerned with Arabic text. We study these algorithms to determine which one is good. We applied the algorithm with five Parts of data and the results showed that the accuracy vary from one algorithm to another depending on the nature and size of data. We can see from the results that the Bayes Net have good accuracy 96.625% when the file size is 600 for each category and we can clearly talk that the highest accuracy is 99.13% which belongs to Random Forest algorithm when the number of file is 1200 files. On the other side when removing stop words, punctuations and diacritics from original data the highest accuracy is 96.73% accuracy achieved by the NB Multinomial with 600 file and the highest accuracy is 98.83% belongs to the Kstar when increase the number of file to 1200, and when applying the light 10 stemmer we got accuracy 96.73% belongs to NB Multinomial algorithm with 600 file. The highest accuracy is 99.51% which belongs to Random Forest algorithm with 1200 file. When applying Chen stemmer we got accuracy 96.02% belongs to NB Multinomial algorithm with 600 file, and when the file become 1200 the 99.51% accuracy belongs to Random Forest algorithm, finally when applying Khuja algorithm for extracting the roots the NB Multinomial and Bayes Net have the same accuracy and NB Multinomial have highest accuracy 95.32% when the number of file was 600, and at the last that the highest accuracy is 99.51% belongs to the Random Forest classifier with 1200 file.

References


