Proc. of the Intl. Conf. on Advances in Computing, Electronics and Communication-- ACEC 2013

ISBN: 978-981-07-7965-8

Arabic Text Categorization using Rocchio Model

[Abduelbaset Goweder[1], Mohummed Elboashi[2], and Kilian Stoffel[3]]

***Abstract*— Automatic text categorization is considered an important application in natural language processing. It is the process of assigning a document to predefined categories based on its content. In this research, some well-known techniques developed for classifying English text are considered to be applied on Arabic. This work focuses on applying the well-known Rocchio (Centroid-based) technique on Arabic documents. This technique uses centroids to define good class boundaries. The centroid of a class c is computed as center of mass of its members. Arabic language is highly inflectional and derivational which makes text processing a complex task. In the proposed work, first Arabic text is preprocessed using tokenization and stemming techniques. Then, the Rocchio Algorithm is adopted and adapted to be applied to classify Arabic documents. The implemented algorithm is evaluated using a corpus containing a set of actual documents. The results show that the adapted Rocchio algorithm is applicable to categorize Arabic text. Ratios of 92.2%, 92.7%, and 92.1% of Micro-averaging recall, precision, and F-measure respectively are achieved, against a data set of 500 Arabic text documents covering five distinct categories.**

***Keywords*—** Rocchio algorithm, Centroid-based Algorithm, Text Mining, Machine Learning, Arabic Text Categorization, Arabic Text Classification.

I. Introduction

Nowadays, great amounts of textual information are available on the Internet. In huge text collections, identifying the relevant document related to a specific topic is really a challenging task. Also, the process of searching for the relevant document is too expensive as it has to search every document in the entire collection and hence results in huge computational as well as time complexity. This leads to the topic problem called Text Categorization (TC), also known as Text Classification. It is the process of assigning tags to documents with one or more predefined classes [18]. Even though assigning classes manually to a given set of documents is accurate, however in huge collections, this manual approach becomes very slow and useless.

[1]Abduelbaset Goweder
Surman Higher Institute
Libya

[2]Mohummed Elboashi
The Libyan Academy
Libya

[3]Kilian Stoffel
Neuchatel University
Switzerland

In order to speed up this process, automatic text categorization was introduced. Automating text categorization helps both in organizing and finding information across the huge collection much easier and faster as well ([23]. TC is based on machine learning and statistical techniques inherited from the IR studies. Such methods are applied to a set of documents (training set) in order to automatically learn the target categorization function [13]. Automatic learning algorithms extract some statistical properties from documents. Such properties are then used to categorize documents. The learning algorithms need to be provided with document representations such as the set of document words which seems to be sufficient to achieve accurate representations ([24]. This representation is very common in IR and it is often referred to as bag-of-words and has shown high accuracy in automated TC [16,17].

Classifying Arabic text differs from classifying English one, because Arabic is highly inflectional and derivational language. Also, some of the vowels of Arabic scripts are represented by diacritics which are usually not written in the text. This leads to the fact that some important information are lost.

On the basis of high performance of the Rocchio technique on English Text [6,11,12] and due to the fact that limited work on automatic Arabic text categorization, we were strongly motivated to adopt and adapt this technique to be applied for categorizing Arabic text. The task can first be accomplished by pre-processing data using NLP techniques such as stop-words removal and stemming. Then, the Rocchio Classification technique is applied to classify Arabic documents. Finally, the performance of Rocchio classification technique on Arabic text will be evaluated.

II. Background

The categorization of digital documents in general categories (e.g., News, Economic, Sports, Religion ,..etc) is an interesting topic to improve the performance of IR systems. The literature reveals that the most (computationally) efficient models are based on a vector representation of both documents and categories by means of feature weights [7]. The decision of a document category is made by measuring the similarity between the target vector pair (i.e., document and category).

Automated, efficient, and accurate TC has a large applicability in the design of IR systems. In the same way, IR is usually exploited for designing NLP applications such as Information Extraction (IE), Question/Answering (Q/A) and Text Summarization (TS).

Proc. of the Intl. Conf. on Advances in Computing, Electronics and Communication-- ACEC 2013

ISBN: 978-981-07-7965-8

A. A Text Categorization System

A text categorization system might consist of a set of components, these are: Feature Extraction, Dimensionality Reduction, Classifier Training, and Thresholding.

The purpose of feature extraction process is to transform a textual document into a list of items (attributes) to be processed by machine learning algorithms. Identification of items as features to represent a document is an important task. Due to language complexity and ambiguity, feature extraction becomes a difficult task. Most common schemes for feature extraction are: Bag of Words, Word n-grams, and TS.

The Bag of Words (BOW) is the most common approach used for feature extraction. So in this paper, we only discuss this approach. In this approach, every word can be considered as an attribute to represent a document. To improve the performance of a categorization system, a proper weighting of a feature can be applied. Once the set of features are selected, they might be weighted according to their relative importance for the document in the collection [20]. Since one of the most common weighting schemes used in the field of text categorization is TF_IDF (Term-Frequency Inverse-Document-Frequency); therefore, this weighting scheme is used in our work. In the TF_IDF scheme, the definition of the term weight can be given by:

$$w_{ij} = f_{ij} \log(N/n_i) \quad (1)$$

Where:

- w_{ij} is the weight of term i in document j.
- f_{ij} is the frequency of term i in document j.
- N is the total documents in the collection.
- n_i is the number of documents in the collection that contain term i.

The second component of a categorization system is the Dimensionality Reduction (DR) which is responsible for decreasing hundreds of thousands of features. This task can be achieved using two main approaches: discard low-importance features, or transform features from one space into another. Low-importance features can be filtered out using a very common factor which is the document frequency. This factor is used to remove terms that are very rare or too frequent. By removing those features appearing in one document and too frequent terms (stop-words), we may reduce considerably and significantly the feature space. Stop-words are meaningless and insignificant terms such as: determiners, prepositions, auxiliaries, etc. Transforming features from one space into another can be accomplished via text summarization. Performing in this manner both feature selection and dimensionality reduction are achieved.

The third component of a categorization system is the Classifier Training. After documents are converted into a list of features, these features are fed to classifiers to be trained [16]. Some of the well-known classifiers are: Probabilistic, Decision Trees, Sample Based, Linear Classifiers, and Centroid-based Document Classifiers. According to Cardoso and Oliveira [6], computationally simple and fast Centroid-based models can achieve high performance compared with other top-performing models. On the basis of this conclusion, Centroid-based models are adopted and discussed in this work. In the Rocchio (Centroid-based) algorithm, the documents are represented using the vector-space model [21]. In this model, each document is represented by the TF_IDF representation vector, *i.e.*, *dtf-idf* = ($tf_1 \log(N/df_1)$, $tf_2 \log(N/df_2)$, . . . , $tf_n \log(N/df_n)$). Construction of the Rocchio model can be done according to one of the following three methods: *Centroid-sum*, *Centroid-average* , and *Centroid-normalizedsum*.

In the *Centroid-sum* method, each class C_j is represented by a vector which is the sum of all document vectors of the positive training instances for this class:

$$\overrightarrow{C_J} = \sum_{d_i \in C_j} \overrightarrow{d_\iota} \quad (2)$$

In the *Centroid-average* method, each class C_j , which has $|C_j|$ documents, is represented by the average of all the vectors of the positive training instances for this class:

$$\overrightarrow{C_J} = \frac{1}{|C_j|} \sum_{d_i \in C_j} \overrightarrow{d_\iota} \quad (3)$$

In the *Centroid-normalizedsum* method, each class C_j is represented by a vector which is the sum of all the vectors of the positive training instances for this class, normalized so that it has unitary length:

$$\overrightarrow{C_J} = \sum_{d_i \in C_j} \frac{\overrightarrow{d_\iota}}{|\overrightarrow{d_i}|} \quad (4)$$

In our work, the *Centroid-normalizedsum* method is adopted for its simplicity and high performance according to Cardoso and Oliveira [6].

After centroids of different categories are identified using one of the *Centroid* methods, a new unseen document can be classified by determining the closest centroid to the document vector. The category of this centroid is then assigned to the tested document. Similarity measures, such as: Cosine similarity, Pearson Correlation Coefficients, and Euclidean-based similarity, can be employed. These similarity measures are used to compute the distance between the tested document vector and the centroid vector. Consequently, the class of the tested document can be determined as follows:

$$C' = arg_j \, max \, (\vec{d}. \overrightarrow{centroid_j}) \quad (5)$$

The fourth component of a categorization system is the Thresholding. Text categorization models perform categorization tasks based on certain values (similarity measures, probabilities, etc.). These values, which are usually determined empirically to provide best performance, should be thresholded to establish the goodness of the assigned class.

B. Text Categorization Evaluation

A text categorization system can be evaluated by conducting a series of experiments to assess its performance. In the test phase, test examples, that have been already labeled by human, are fed to the system to be classified. The evaluation process can be started by matching human-labeled categories with machine-labeled ones. Four possible situations of categorization can be summarized in the following contingency table as presented in Table I.

Proc. of the Intl. Conf. on Advances in Computing, Electronics and Communication-- ACEC 2013

ISBN: 978-981-07-7965-8

TABLE I. A CONTINGENCY TABLE.

Class C_i		**Assigned by a human-expert?**	
		Yes is correct	***No is correct***
Assigned by a Classifier?	***Predicted Yes***	*TPi*	*FPi*
	Predicted No	*FNi*	*TNi*

Where:

TPi - True Positive: the number of cases correctly assigned by both a classifier and human-expert.

TNi - True Negative: the number of cases correctly discarded by both a classifier and human-expert.

FPi - False Positive: the number of cases incorrectly assigned by a classifier.

FNi - False Negative: the number of cases incorrectly discarded by a classifier.

Class C_i - indicates that those values in the contingency table are computed for every class.

Some well-known measures used to gauge a system's performance can be computed as follows:

Recall is defined as the fraction of target (human-expert) labels that a classifier found, i.e.,

$$\text{Recall (R)} = TP/(TP+FN) \quad (6)$$

Precision is defined as the fraction of assigned labels that that a classifier got right, i.e.,

$$\text{Precision (P)} = TP/(TP+FP) \quad (7)$$

In some situations, the balance between Recall and Precision measures is a difficult task, another criterion, which is F-measure, could be used as an overall performance measure. F-measure combines Recall and Precision, and is expressed by the following equation:

$$\text{F-measure (F)} = (2PR)/(P+R) \quad (8)$$

It is also possible to evaluate a system's performance by making use of the accuracy and error rates to gauge the percentage of correct and wrong cases respectively. These measures can be computed according to the following formulae:

$$\text{Accuracy (A)} = (TP+TN)/(TP+FP+FN+TN) \quad (9)$$

$$\text{Error (E)} = (FP+FN)/(TP+FP+FN+TN) \quad (10)$$

Previous system's performance measures provide indications for a single class c_i. To obtain global indications, two different averaging approaches can be used: macro-averaging and micro-averaging [15]. Since micro-averaging seems to be the preferable averaging method in the literature [23], therefore; micro-averaging is adopted in this work. Micro-averaging generates a global contingency table as shown in Table II.

TABLE II. A GLOBAL CONTINGENCY TABLE.

Category set $C = \{c_1, c_2, \ldots, c_{|C|}\}$		**Assigned by a human-expert?**	
		Yes is correct	***No is correct***
Assigned by a Classifier?	Predicted Yes	$\sum_{i=1}^{\lvert C\rvert} TPi$	$\sum_{i=1}^{\lvert C\rvert} FPi$
	Predicted No	$\sum_{i=1}^{\lvert C\rvert} FNi$	$\sum_{i=1}^{\lvert C\rvert} TNi$

Then, a single effectiveness measure is computed by summing over all individual decisions. For example, micro-averaging recall (MI-Recall) is computed based on the global contingency table as follows:

$$\text{MI - Recall} = \frac{\sum_{i=1,|c|} TPi}{\sum_{i=1,|c|} TPi + FNi} \quad (11)$$

III. Literature Review

Automatic categorization research on documents written in European languages such as English, German, Italian and Spanish has been intensively carried out. In addition, work on Asian languages such as Chinese and Japanese is also obtained a high consideration. However, research on automatic Arabic text categorization has been paid less attention compared with the work done on European languages. Classifying Arabic text is different from classifying English one, because Arabic is highly inflectional and derivational language [2]. Also, in Arabic scripts, some of the vowels are represented by diacritics that are usually unwritten. Consequently, some information will be lost. In this section, some work on categorization of English text is reviewed. Additionally, some studies on Arabic text categorization are discussed.

A. *English Text Categorization Work*

Cardoso and Oliveira [6] experimentally evaluated several centroid based models on text categorization tasks. They show that: (1) the TF_IDF term weighting scheme is very effective compared with recent approaches. (2) determining the centroid of a class using the Centroid-NormalizedSum model always outperforms other models. (3) the Centroid-based model produces results roughly similar to the top-performing support vector machines (SVM) model. Based on these results, the Centroid-NormalizedSum model is adopted to be implemented in our work.

Han and Karypis, [12] worked on document categorization using the Centroid-based model. They compared the performance of the Centroid-based classifier with that of other classifiers such as Naïve Bayesian (NB), Decision Tree (C4.5), and k-Nearest-Neighbor (KNN) on a variety of document collections. Their results reveal that the Centroid-based classifier considerably outperforms others.

Guan et al. [11] designed a fast Class-Feature-Centroid (CFC) classifier. In CFC, a centroid is built from. CFC proposes a novel combination of two important class distributions: inter-class term index and inner-class term index and employs a denormalized cosine measure to compute the similarity coefficient between a text vector and a centroid. Experiments on the Reuters-21578 corpus and 20-newsgroup email collection show that both Micro-F and Macro-F values are above 0.99, and 0.92 respectively.

B. *Arabic Text Categorization Work*

For heavily inflectional languages such as Arabic, text pre-processing is an essential stage in text categorization particularly and text mining generally. Said et al. [19]

Proc. of the Intl. Conf. on Advances in Computing, Electronics and Communication-- ACEC 2013

ISBN: 978-981-07-7965-8

evaluated several morphological tools for Arabic text categorization. The study examines the performance of Arabic text categorization using the raw text, the stemmed text, and the root text. The results illustrate that light-stemming pre-processing combined with a good performing feature selection method improve the performance of Arabic text categorization. Based on these observations, light-stemming is adopted as a pre-processing technique in our research.

Sawaf et al. [22] conducted experiments on the Arabic NEWSWIRE corpus using statistical methods without morphological analysis. Their text categorization system is based on the maximum entropy technique. Their work is evaluated using the most common criteria: Precision, Recall, and F-measure. In their experiments, no pre-processing on text was carried out.. The results show that the best categorization accuracy was 62.7% with precision of about 50%.

EL Kourdi et al. [9] automatically classified Arabic web documents using NB Algorithm. Prior to categorization, Arabic documents are typically pre-processed: stop-words are removed, vowels stripped off, and roots extracted. Their results show that the overall average accuracy was about 68.78%.

Bawaneh et al. [5] implemented NB and KNN algorithms to automatically classify Arabic documents. In their study, the evaluation process of their systems' performance is carried out on a data set consisting of 242 documents that belong to 6 categories. The k-fold cross-validation method is used to assess the accuracy. For the implemented NB and KNN algorithms, the overall average accuracy was about 73.6% and 84.2 respectively.

El-Halees et al. [8] built ArabCat system that classifies Arabic documents using Maximum Entropy method. In their work, pre-processing techniques such as tokenization and stemming are first applied on data sets collected from the Web. Then, the data sets are classified using maximum entropy method. The performance of their system achieved 80.48% of recall, 80.34% of precision, and 80.41% of F-measure.

Al-Shalabi et al. [4] automatically classified Arabic documents using KNN algorithm. In their work, feature extraction and reduction was implemented using the Document Frequency (DF) threshold method. Experiments were conducted on a data set of 621 Arabic text documents that belong to 6 different categories for training and testing. The results show that the KNN algorithm is applicable to Arabic text. They recorded micro-average precision and recall scores of about 0.95.

Alsaleem [3] investigated the performance of two well-known algorithms: NB and SVM on different Arabic data sets. The results indicate that the SVM algorithm outperformed NB algorithm with regard to Recall, Precision, and F-measure.

Abidi et al. [1] carried out a comparative study to assess the effect of a conceptual representation of the text. They implemented an Arabic classifier using KNN algorithm. In their work, feature extraction was achieved by applying three different schemes, these are: Bag of Words, N-grams, and a conceptual representation. They evaluated the KNN algorithm using these schemes. The following figures of F-measure were achieved: 64%, 68%, and 74% for Bag of Words, N-grams, and a conceptual representation schemes respectively. Their results show that the conceptual representation scheme outperforms the other two schemes.

IV. Arabic Categorization System

In this section, an Arabic text categorization system is designed and implemented based on the well-known Rocchio categorization algorithm. This algorithm has been adopted and adapted to classify Arabic text into pre-defined categories. The proposed system is designed by building a training model to train the system, and a testing model to evaluate the system's performance.

In the Rocchio categorization algorithm, the documents are represented using the vector-space model. In this model, each document d is represented by feature weight vector [12]. For each set of documents belonging to a particular class, their centroid vectors are determined. If number of classes is k, this produces k centroid vectors: $\{\overrightarrow{C_1}, \overrightarrow{C_2}, \overrightarrow{C_3}, \ldots\ldots, \overrightarrow{C_t}\}$, where each $\overrightarrow{C_t}$ is the centroid for the i^{th} class. The category of a previously unseen document *x* is identified as follows: first, the document-frequencies of the various terms extracted from the training set are used to compute the TF_IDF weighted vector-space representation of document *x*. Then, the similarity between document *x* and k centroids is computed using the Cosine measure. Finally, based on these similarities, the document *x* is assigned to the class corresponding to the closest centroid. The following sub-sections describe training and testing models which were designed and used in the proposed system.

A. *The Training Model*

In this model, text pre-processing procedure is first applied on the training documents to extract the features. Then, the extracted features are weighted using TF-IDF weighing scheme. The features extraction is a key component used to extract keywords from a raw text. The features extraction can be achieved by applying the following procedure (referred to as the features extraction algorithm) which includes seven main steps:

1- Read the raw text file that is used to train the system.
2- Convert the encoding scheme of the text file into UTF-8 encoding.
3- Remove the punctuations, non-Arabic letters, and all special characters from the text.
4- Normalize some Arabic letters.
5- Tokenize the text file to obtain a set of tokens.
6- Remove all Arabic stop-words from the text.
7- Stem text to obtain a base form of a word.

Step 4 in the above procedure is concerned with Arabic alphabet normalization. Arabic letter normalization is adopted from Larkey et al. [14] and implemented in the training model. In this process, some letters are eliminated or replaced by other letters. As an example, omit the letter *Hamza* (ء) from

Proc. of the Intl. Conf. on Advances in Computing, Electronics and Communication-- ACEC 2013

ISBN: 978-981-07-7965-8

Alef (ا). As a second example, replace the final letter *Alef-maqsoura* (ى) with *Yeh* (ي), and *Teh-marbuta* (ة) with *Heh* (ه), etc…

Step 6 is concerned with stop-words removal. As mentioned earlier in section II(*A*), stop-words are insignificant and appear very frequent in a text. So, removing them reduces the space of the items significantly. Based on this conclusion, stop-words list constructed by Goweder [10] was adopted and implemented in our work. Step 7 is concerned with text stemming. As mentioned in section III(*B*), studies illustrate that light-stemming pre-processing technique improves the performance of Arabic text categorization. Based on this observation, the light-stemming "light8" algorithm developed by Larkey et al. [14] is adopted and implemented in our research.

Following steps of the above procedure yields a set of features that describe each document. Then, a database table is built for these features to be weighed using the TF-IDF weighing scheme as discussed in section II(*A*). Now, each document in the training set is represented as a vector. After document vectors are constructed, the *Centroid-normalizedsum* method is adopted and implemented for its simplicity and high *performance* according to Cardoso and Oliveira [6]. In this method, each class is represented by a vector which is the sum of all the vectors for the positive training examples for this class, normalized so that it has unitary length as given in equation (4) in section II (*A*). The main steps of the implemented training model can be summarized as illustrated in Fig. 1.

B. *The Testing Model*

After the centroid vectors are determined , the test document vector is built using the same features extraction algorithm used for the training model and described in section IV(*A*). Then, the extracted features are weighed using the same weighing scheme (TF-IDF) used for the training model and discussed in section II(*A*).

Now, we obtained a test document vector and k centroid vectors. The similarity coefficient between the test document vector and k centroid vectors is computed using the following Cosine measure:

$$\text{Cos}(\vec{d}, \vec{C}) = \frac{\vec{d}.\vec{c}}{|\vec{d}||\vec{c}|} \quad (12)$$

where the numerator represents the *dot product* of the vectors $\vec{d}$ and $\vec{C}$, while the denominator is the product of their *Euclidean lengths*.

Finally, based on these similarities, the class of the tested document is identified according to the most similar centroid. That is, the class of the tested document can be determined according to equation (5) given in section II(*A*).

The main steps of the implemented testing model can be summarized as illustrated in Fig. 2.

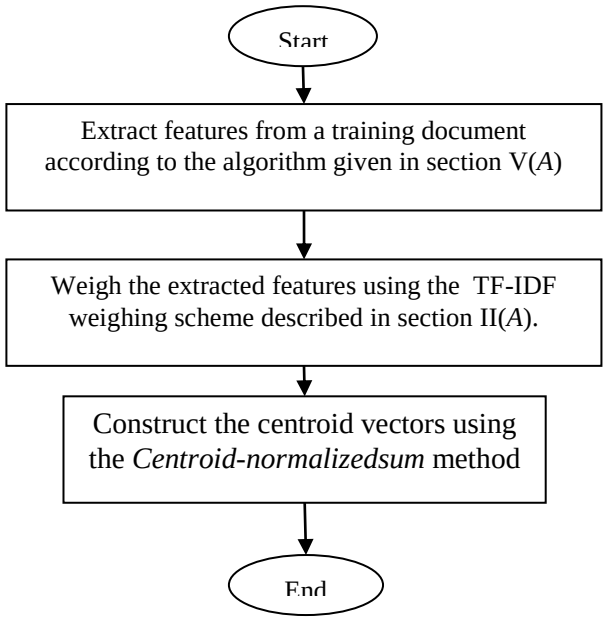

Figure 1: The Training Model Flowchart.

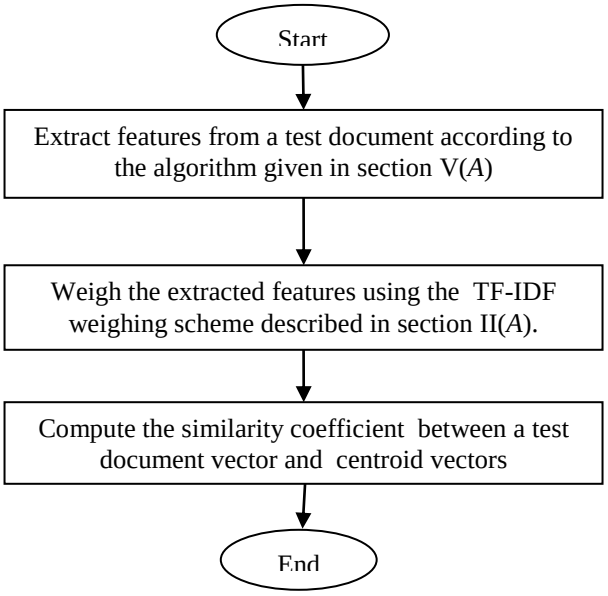

Figure 2: The Testing Model Flowchart.

V. Experiments and Evaluation

This section presents the set of experiments which was conducted, the evaluation method used to assess the overall performance of the implemented Arabic categorization system, and the results discussion.

A. *Experiments*

The corpus (dataset) used to train and test the implemented classifier consisted of a set of articles from different domains. These articles are collected from Arabic Web-sites: Al-Nahar, Al-Dostor, and Aljazeera (the Qatari television news channel in Arabic). Five hundred (500) documents are gathered

Proc. of the Intl. Conf. on Advances in Computing, Electronics and Communication-- ACEC 2013

ISBN: 978-981-07-7965-8

covering five (5) subject categories, these are: politics, culture & arts, business & economy, sports, and medical & health. These categories are the most commonly used in Arabic NLP. Each subject category consists of a hundred (100) documents. Four hundred (400) documents were used to train the classifier, whereas one hundred (100) documents were used to test the classifier. The documents assigned to the training and testing sets were randomly chosen.

The collection of documents was split using the k-fold cross validation method which partitions the collection into k different equally sized sets: (S_1, S_2, … ,S_k). In this partitioning, positive and negative cases for each category are equally distributed. The k-fold cross-validation was adopted in this work because it minimizes variations due to the biased sampling of training data. This method is not widely used in many Arabic NLP studies. In our case, five datasets were created. Each contains 100 documents. In each experiment, four sets will be used for training (i.e., 400 documents), and one for testing (i.e.,100 documents).

In our work, five experiments were iteratively conducted by applying the train-and-test approach on five train-test sets. The overall performance of the implemented classifier is determined by computing the average of the five runs. Section V(*C*) discusses these experiments and their results.

B. *Evaluation Approach*

The evaluation process of the classifier can be undertaken by determining the performance of the implemented classifier on real data (not machine generated). To measure a performance, effectiveness is a common evaluation method which refers to the ability to take the right decision on the categorization of unseen documents. There are several commonly used performance measures of effectiveness such as recall, precision, etc… as discussed in section II(*B*). Additionally; Micro-averaging, described in section II(*B*), is widely used as a global measure for the performance.

For each experiment, a four cell contingency table is generated for each category as discussed in section II(*B*) and depicted in Table I. The conventional performance measures of effectiveness are computed from this contingency table according to the formulae given in section II(*B*). To assess the overall performance for each experiment, a global contingency table is created as depicted in Table II, section II(*B*). Referring to Table II, Micro-averaging measures such as "MI-Recall", MI-Precision, etc…were defined and computed.

C. *Discussion of the Results*

In this section, the results of different experiments are presented and discussed. In the first experiment, contingency tables are generated for all categories. As a sample, we have only chosen the contingency table for culture&arts category to be discussed. Table III illustrates this contingency table. This table reveals that 17 documents were correctly assigned to culture&arts category, and 78 documents were correctly discarded from culture&arts. In addition, two documents were incorrectly assigned to culture&arts, and three documents were incorrectly discarded from culture&arts. Based on these results and those of other categories, the performance measures are computed. The results of the performance measures of all categories are shown in Table IV. It is noticeable that the best performance is achieved by the classifier on Medical & health domain. On the other hand, the lowest performance is recorded on Politics domain. For better illustration, the results shown in Table IV are plotted as histograms as shown in Fig. 3.

TABLE III. A CONTINGENCY TABLE FOR CULTURE & ARTS.

Culture & Arts category		**Documents that**	
Documents that are		***belong to this category***	***do not belong to this category***
	assigned to this category	*17*	*2*
	not assigned to this category	*3*	*78*

TABLE IV. FIRST EXPERIMENT'S PERFORMANCE.

Category	**Evaluation Criteria**				
	R%	P%	F%	A%	E%
Politics	95	76	84.4	93	7
Culture	85	89.5	87.2	95	5
Business	75	100	85.7	95	5
Sports	95	90.5	92.7	97	3
Medical	100	100	100	100	0

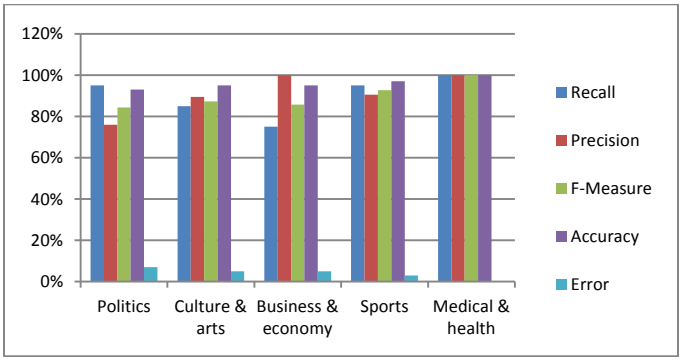

Figure 3. First experiment's performance measures.

For the first experiment, a global contingency table is generated in order to compute Micro-averaging measures. Table V shows the results of a global contingency table. Based on the values shown in Table V, Micro-averaging measures were computed and the results are shown in Table VI. Looking at these results, it is observable that the classifier is recording high performance. Fig. 4 shows the histogram of these results.

Proc. of the Intl. Conf. on Advances in Computing, Electronics and Communication-- ACEC 2013

ISBN: 978-981-07-7965-8

TABLE V. A GLOBAL CONTINGENCY TABLE FOR SET C.

Category set C		Documents that	
		belong to C	*do not belong to C*
Documents that are	*assigned to C*	*90*	*10*
	not assigned to C	*10*	*390*

TABLE VI. MICRO-AVERAGING PERFORMANCE MEASURES.

	Evaluation Criteria				
	R%	P%	F%	A%	E%
Micro-averaging	90	90	90	96	4

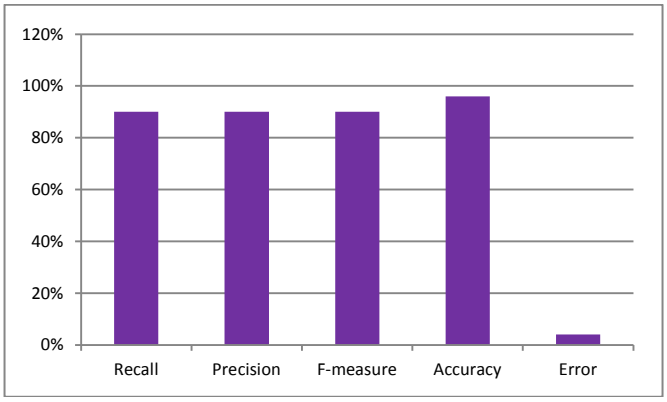

Figure 4. Micro-averaging performance measures.

Due to the limited writing space, the results of the rest of experiments are not presented in this section. However, the results of all experiments were averaged to draw a conclusion about the average performance of the implemented classifier. The average performance is determined by computing the average of the results of all experiments. Table IIV shows the average performance measures of all categories for the five experiments. These results suggest that Medical & health category outperformed other categories. Whereas Politics domain achieved the poorest performance. The histogram of these results is given in Fig. 5.

Finally, the overall performance is determined by computing Micro-averaging measures of five experiments. The results are shown in Table VII and Fig. 6. These results reveal that the overall performance of the implemented classifier is quite high and can be considered very promising.

TABLE IIV. THE AVERAGE PERFORMANCE OF FIVE EXPERIMENTS.

Category	Evaluation Criteria				
	R%	*P%*	*F%*	*A%*	*E%*
Politics	93	83.4	87.7	94.8	5.2
Culture	90	91.8	90.7	96.4	3.6
Business	85	93.8	88.8	95.8	4.2
Sports	93	97.1	94.7	98	2
Medical	100	97.2	98.5	99.4	0.6

.

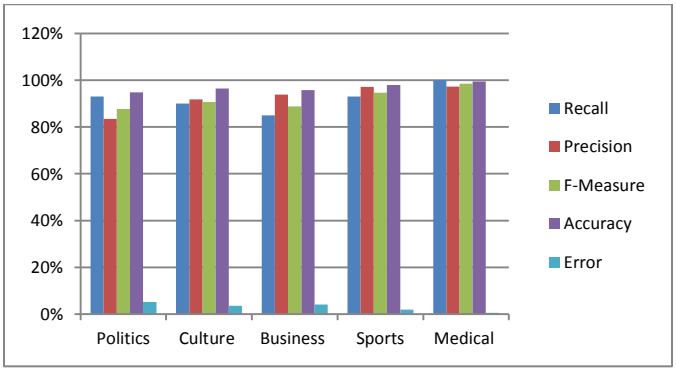

Figure 5. The average performance of 5 experiments.

TABLE VII. MICRO-AVERAGING OVERALL PERFORMANCE.

	Evaluation Criteria				
	R%	*P%*	*F%*	*A%*	*E%*
Micro-averaging	92.2	92.7	92.1	96.9	3.1

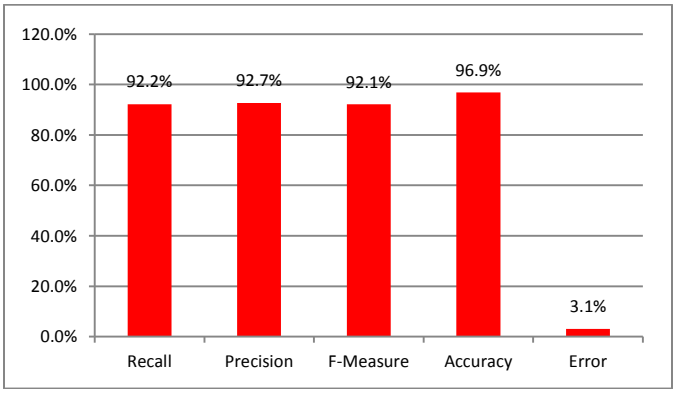

Figure 6. Micro-averaging Overall Performance.

Proc. of the Intl. Conf. on Advances in Computing, Electronics and Communication-- ACEC 2013

ISBN: 978-981-07-7965-8

VI. Conclusions and Future Work

In this paper, an Arabic classifier was designed and implemented based on the well-known Rocchio (Centroid-based) algorithm. An evaluation of the performance of this classifier was carried out. The effectiveness of categorization was measured by applying the conventional evaluation criteria for local evaluation and the Micro-averaging for global evaluation. The obtained results show that high Micro-averaging scores of recall (92.2%), precision (92.7%), and F-measure (92.1%) are achieved using a corpus of 500 Arabic text documents covering five different categories. It can be pointed out that the very satisfactory results suggest the applicability of Rocchio algorithm on Arabic text. It is also observable that our results are comparable with the best existing results in this field. Our results can be considered excellent and very promising. Finally, it can be concluded that the implemented Arabic classifier has achieved high figures of evaluation criteria and is considered among the best performing classifiers.

Our future work focuses on scaling up the corpus used for training and testing the classifier. In addition, we try to exclude noisy training examples by finding a threshold point which eventually improves the whole performance.

Acknowledgment

We would like to express our gratitude to the Libyan Ministry of Higher Education and Scientific Research for supporting this work.

References

[1] Karima Abidi et al. "*Arabic Text Categorization: A Comparative Study of Different Representation Modes.*" Journal of Theoretical and Applied Information Technology, 2012,38(1).

[2] Hammo Abu-Salem et al. "*A Question Answering System to Support the Arabic Language.* " ACL 2002, Philadelphia, PA, 2002, p 55-65.

[3] Saleh Alsaleem. "*Automated Arabic Text Categorization using SVM and NB.*" International Arab Journal of e-Technology, 2011, 2(2).

[4] Riyad Al-Shalabi et al. "*Arabic Text Categorization using KNN Algorithm.*" Amman Al-Ahliya University, Jordan, 2008.

[5] Mohammed J. Bawaneh, et al. "*Arabic Text Classification using K-NN and Naive Bayes.*" Al-Balqa Applied University, Journal of Computer Science, 2008, 4 (7): 600-605.

[6] Ana Cardoso-Cachopo and Arlindo L. Oliveira. "*Empirical Evaluation of Centroid-based Models for Single-label Text Categorization*." INESC-ID Technical Report 7/2006.

[7] Ido Dagan et al. "*Mistake-driven learning in text categorization.*" 2nd Conference on EMNLP, 1997.

[8] Alaa M. El-Halees. "*Arabic Text Classification Using Maximum Entropy.*" The Islamic University of Gaza, Journal of Computer Science, 2007, 15(1): 157-167.

[9] M. El-Kourdi et al. "*Automatic Arabic Document Categorization Based on the Naïve Bayes Algorithm.*" 20th International Conference on Computational Linguistics. Geneva, 2004.

[10] Abduelbaset Goweder. "*Arabic Stemming and Information Retrieval: The Case of Broken Plurals.*" PhD Thesis, University of Essex, Colchester, England, 2004.

[11] Hu Guan, Jingyu Zhou, Minyi Guo. "*A Class-Feature-Centroid Classifier for Text Categorization.*" Shanghai Jiao Tong University, Department of Computer Science, 2009.

[12] Eui-Hong Han and George Karypis. "*Centroid-Based Document Classification: Analysis & Experimental Results.*" University of Minnesota, Department of Computer Science, 2000.

[13] M. Ikonomakis et al. "*Text Classification Using Machine Learning Techniques.*" University of Patras, Greece, 2005.

[14] L. Larkey et al. "*Improving Stemming for Arabic Information Retrieval: Light Stemming and Co-occurrence Analysis.*" SIGIR'02, Tampere, Finland, 2002.

[15] D. Lewis. "*Representation and Learning in Information Retrieval.*" PhD thesis, Department of Computer Science, University of Massachusetts, 1992.

[16] Arturo Montejo-Raez. "*Automatic Text Categorization of documents in the High Energy Physics domain*." 2005.

[17] Alessandro Moschitti. "*Natural Language Processing and Automated Text Categorization*." PhD thesis, University of Rome Tor Vergata, 2003.

[18] Rajendra Prasath. "*Enhancing Text Categorization Using External Knowledge Repositories*". Master thesis, Indian Institute of Technology, 2008.

[19] Dina Said et al. "*A Study of Text Preprocessing Tools for Arabic Text Categorization.*" Informatics Department, Cairo University, Egypt, 2007.

[20] Gerard Salton et al. "*A Vector Space Model for Automatic Indexing.*" Technical Report TR74-218, Cornell University, 1974.

[21] Gerard Salton. "*Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer.*" Addison-Wesley, 1989.

[22] H. Sawaf et al. "*Statistical Classification Methods for Arabic News Articles.*" ACL 2001, Toulouse, France, 2001.

[23] Fabrizio Sebastiani. "*Machine learning in automated text categorization*." ACM Computing Surveys, 2002, 34(1):1–47.

[24] Y. Yang and X. Liu. "*A re-examination of text categorization methods*." ACM SIGIR Conference, 1999.