Classifying Arabic Text Using KNN Classifier

Amer Al-Badarenah
Computer Information Systems Department
Jordan University of Science and Technology, Irbid, Jordan

Emad Al-Shawakfa
Computer Information Systems Department
Yarmouk University, Irbid, Jordan

Khaleel Al-Rababah
Computer Science Department
University of New Brunswick
Canada

Safwan Shatnawi
Applied Studies College
University of Bahrain
Bahrain

Basel Bani-Ismail
Department of Computer Science
Sultan Qaboos University
Oman

***Abstract*—With the tremendous amount of electronic documents available, there is a great need to classify documents automatically. Classification is the task of assigning objects (images, text documents, etc.) to one of several predefined categories. The selection of important terms is vital to classifier performance, feature set reduction techniques such as stop word removal, stemming and term threshold were used in this paper. Three term-selection techniques are used on a corpus of 1000 documents that fall in five categories. A comparison study is performed to find the effect of using full-word, stem, and the root term indexing methods. K-nearest – neighbors classifiers used in this study. The averages of all folds for Recall, Precision, Fallout, and Error-Rate were calculated. The results of the experiments carried out on the dataset show the importance of using k-fold testing since it presents the variations of averages of recall, precision, fallout, and error rate for each category over the 10-fold.**

Keywords—categorization; Arabic; KNN; stemming; cross validation

I. Introduction

Due to the advances in technology, a huge number of structured and unstructured of text documents is being published online every day. Internet users are interested in reading newspapers online, sending and reading email, participate in chat rooms and blogs, wikis, news groups, and many more. This growing amount of text on the web makes it urgent to automatically structure and categorize this text [1]. Organizations today are faced with a huge volume of information stored in digital form. Much of this information is stored in different types of documents. The increasing availability of documents in digital information has led to a huge interest in categorizing (classifying) documents (TC) [3]. As a result, computer systems are developed to automatically organize and classify documents.

In order to make use of the huge information; information needs to be managed. The end goal of information management is to locate only the relevant documents; a task which requires documents to be categorized and instead of manually classifying documents; a high precision method that performs automatic text categorization is, on the other hand, apparent.

The objective of document classification is to minimize the detail and diversity of the information by grouping similar documents together. Text classification is a process of structuring a set of documents according to a group of structure which is known in advance [1]. Another definition is "document categorization is the process of assigning a text document to one or more predefined categories (labels) based on its content" [4].

Text categorization has many applications such as document routing, document management, documents organization, text filtering, spam filtering, mails routing, word sense disambiguation, news monitoring automatic documents indexing and hierarchal catalogue of web resources. As mentioned above, text filtering is one of the applications of text categorization. Text filtering can be considered as a case of single-label TC that is categorizing of incoming documents into two disjoint categories, the relevant and the irrelevant [6, 7].

Most of the text categorization systems have been developed for English language and just few of the developed systems were for Arabic language [8]. The reason behind having fewer systems developed for Arabic Text Categorization is because of the complex nature of the Arabic Language. The focus of this study is on Arabic Text Categorization (ATC). There are several techniques and algorithms used for text classification such as: Support Vector Machine (SVM), K-nearest Neighbor (KNN), Artificial Neural Networks, Naïve Bayes classifier, and Decision Trees.

This paper is organized as follows: Section 2; describe related works in the area of automatic text categorization. Section 3 describes the Arabic language features and challenges. In section 4, the architecture of text categorization is discussed. Section 5 discusses the used classifying methodology. In section 6, experiments and results are presented. Section 7 shows the conclusions and future work.

II. Literature Review

Many machine learning algorithms have been used in text categorization, those algorithms include: decision tree learning and Bayesian learning, nearest neighbor learning, and artificial

neural networks. A survey presented in [2] discusses the main approaches to text categorization.

The work of [7] showed that applying the KNN classifier using N-Grams and then by using bag of words show that using N-Grams produces better accuracy than using single terms for indexing. In a work presented in [3], a machine learning approach for classifying Arabic text documents is presented; each document was mapped by locating the N-gram frequency technique; the classification was achieved by computing a dissimilarity measure, called the Manhattan distance, between the profile of the instance to be classified and the profiles of all the instances in the training set.

The authors of [4] used three classifiers and compared their performances; the three used classifiers were naïve Bays, k-nearest-neighbors (KNN), and distance-based classifiers. Another work conducted a comparative study of two machine learning methods k nearest neighbor (KNN) and support vector machines (SVM) [9]. Full-word features was used and *tf.idf* as the weighting method for feature selection. The results showed that both methods were of high performance and that SVM showed a better micro average F1 and prediction time.

An intelligent Arabic text categorization was presented in [8], k-nearest neighbor and Rocchio classifiers were used; different term weighting schemes were used also light stemming was used as well. Their results show that Rocchio classifier performs better than k-nearest neighbor classifier. Another study conducted in [10] used stemming and light stemming techniques as feature selection techniques, K-nearest neighbors (KNN) as a classifier. Results reported indicated that light stem was superior over stemming in terms of classifier accuracy. The author of [11] proposed a distance-based classifier for categorizing Arabic text. Each category is represented as a vector of words in an m-dimensional space, and documents are categorized based on their closeness to feature vectors of categories.

III. The Arabic Language Features and Challenges

Arabic language is spoken by more than 250 million Arabic people around the world. In addition, as it is the language of the Holy Quran, Arabic language is understood by more than one billion other Muslims [12]. Arabic alphabet consists of 28 characters:

ا ب ت ث ج ح خ د ذ ر ز س ش ص ض ط ظ ع غ ف ق
ك ل م ن ه و ي

It was indicated by [13] that Arabic language poses various challenges in terms of the language stylistic properties and rules. For example, the authors of [14] show the effect of not using capital letters in Arabic words, which makes it hard to identify proper names, abbreviations and as a result it would makes it complicated in tasks such as in Information Extraction and Named Entity Recognition.

A. Arabic Characters Styles

Arabic characters have different styles when appearing in a word depending on the location of the character in the word whether it is located at the beginning, middle, or end of word and also whether the character can be connected to its neighbor characters or not. For example, the character (س) has different styles according to the location rule, (سـ) if it is located at the beginning of a word as in the word ساعة. It appears as (ـسـ) if the character appears in the middle of a word such as يسهل; (ـس) if the character appears at the end of word as in حبس. Finally, the character (س) will show as (ـس) if it appears at the end of a word but it will not be connected to the character to its right such as in word درس [4].

B. Arabic Diacritics

Diacritics are a property of the Arabic language; it is signals placed below or above letters in order to double the letter when it is pronounced or it acts as a short vowel. Arabic diacritics include: shada, dama, fathah, kasra, sukon, double dama, double fathah, double kasra [4]. It was noted that the absence of the diacritics can lead to a confusing and different meaning. For example, it would be impossible to distinguish between the words حُب which means love and pronounced as hubb and the word (حَب) which means seed and pronounced as habb. So, not having diacritics in most of the modern standard Arabic is considered to be a major challenge to many of Arabic Natural Language Processing (NLP) tasks [13].

C. Arabic Morphology and Word Formation

Arabic language is considered to be a highly inflected language, so it has much richer morphology than English language. For example, Arabic nouns have two genders, feminine and masculine; nouns also can be characterized as singular, dual, or plural. A noun has the nominative case when it is subject; accusative when it is the object of a verb, and the genitive when it is the object of a preposition.

In linguistics, word formation is considered to be a function of morphology. Morphological analysis of human languages is largely based on the following linguistic elements: root, stem, affixes (prefixes, infixes and suffixes), and morphemes [17]. A verb in the Arabic language can be augmented by adding prefixes, infixes and suffixes to refer to the time the event has occurred, whether the verb is plural or singular, and the sex of the participants in the verb. For example the word (أكل), which corresponds to the English verb eat, this verb can have several patterns, for example, if the prefix, characters attached at the beginning of a word, (ي) added to the verb, it becomes (يأكل) which indicates the time of the verb is in present and it is done by one male. On the other hand, if the suffix, a character attached at the end of the word, (ا) added to the verb, the verb becomes (أكلا) which indicates that the time of the event is in the past and it is done by two males.

Table I shows the different derivations for the root word kataba (كتب), its pattern, its pronunciation and the translation of the word in English to show the effect of different form of the word on the meaning [8]. Table II shows different affixes that may be added to the word معلم (Teacher) along with its meaning in English, Gender, and number [8]. Table III shows prefix particle combinations [17].

TABLE I. DIFFERENT DERIVATIONS FOR THE ROOT WORD (كتب)

Arabic Word	Pattern	Pronunciation	English Meaning
كتب	*fa'ala* (فعل)	*kataba*	*wrote*
كتابة	*fe'alah* (فعالة)	*ketaba*	*writing*
كاتب	*fa'el* (فاعل)	*kateb*	*writter*
مكتوب	*maf'ool* (مفعول)	*maktoob*	*is writter*
كتاب	*fe'aal* (فِعال)	*ktaab*	*book*
مكتبة	*maf'alah* (مفعلة)	*maktabah*	*library*
مكتب	*maf'al* (مفعل)	*maktab*	*Office*

TABLE II. DIFFERENT AFFIXES OF TO THE WORD (معلم)

Arabic Word	English Meaning	Gender	Number
معلم	Teacher	masculine	singlular
معلمة	Teacher	feminine	singlular
معلمان	two teachers	masculine	dual
معلمون	teachers	masculine	plural (accusative, genitive)
معلمين	teachers	masculine	plural (nominative)
معلمات	teachers	feminine	plural
المعلم	the teacher	masculine	singlular
والمعلم	and the teacher	masculine	singlular
كالمعلم	as the teacher	masculine	singlular
معلمي	my teacher	masculine	singlular
معلمه	his teacher	masculine	singlular
معلمها	her teacher	masculine	singlular
معلمهم	their teacher	masculine	singlular

TABLE III. PREFIX PARTICLE COMBINATIONS

Combination	Meaning	Example
بال	in the	بالشارع (in the street)
فال	and the, therefore the	فالمدينه (therefore the city)
كال	like the	كالرئيس (like the president)
لال	for the, to the	للمجال (to the field)
وال	and the	والجامعه (and the university)
فبال	therefore in the	فبالحق (therefore in the right)
وبال	and in the	وبالوسط (and in the center)
وكال	and like the	وكالشمس (and like the sun)
ولال	and for the	ولليسار (and for the left)
فب	and in, therefore in	فبنوم (therefore in sleep)
وب	and in	وبحركه (and in movement)
فل	and for, therefore to	فلمعركه (and for battle)
ول	and for, and to	ولزمان (and to time)

IV. ARCHITECTURE OF TEXT CATEGORIZATION

The text categorization (TC) process consists of three key components: data pre-processing, classifier construction, and document categorization, as shown in Figure 1. Data pre-processing implements the function of transferring the original document into a compact representation and will be uniformly applied to training, validation, and classification phases. Classifier construction does inductive learning from a training set of documents, and document categorization process is document classification. In Fig. 1, the arrow with dashed line represents the data flow in the categorization process and the arrow with the solid line represents the data flow in the classifier construction process.

A. Data Pre-Processing

Text documents consist of words made of characters, digits, and special symbols. The pre-processing phase focuses on extracting the words which best describing the document and eliminate the others. This all can be done through many steps such as normalization, dimensionality reduction, and feature creation [15].

B. Normalization

Normalization is the process of finding the standard form for all words found in the documents of the corpus [11]. The normalization process consists of the following steps:

1) Punctuation marks removal

2) Stop words removal, stop words are useless words; stop words include: prepositions, definition articles, and conjunctions.

3) Non-letters removal

4) Diacritics removal

5) Replace initial إ or أ with bare alif ا, replace آ with ا, replace the sequence ىء with ئ, replace final ى with ي, and replace final ة with ه

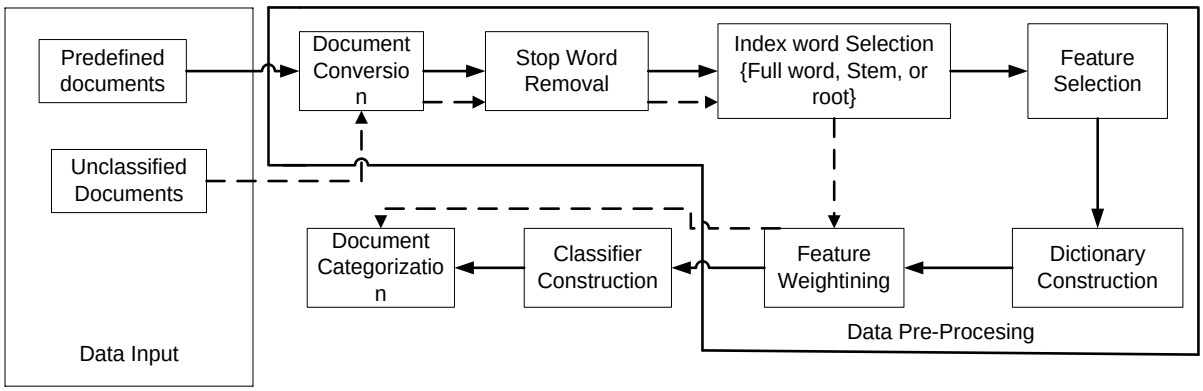

Fig. 1. The architecture of text categorization

C. Feature Selection and Reduction

A text document can have a large number of features (words). Imagine the case where you have thousands of text documents and each document is represented by a vector; vector entries are the frequencies with which each word occurs in the document. There are many gains to dimensionality reduction [15]:

1) Many data mining algorithms perform better if the dimensionality – the number of attributes in the document is lower; the reason for this benefit is that because the dimensionality reduction can eliminate unsuitable features.

2) Dimensionality reduction can lead to a more understandable model because the model may involve fewer attributes.

3) Dimensionality reduction will facilitate data visualization.

The following are two techniques for feature set reduction:

1) Feature Selection. Document vector dimensionality can be reduced by selecting just a subset of original features. The objective of this phase is to eliminate the features (words), which can be considered to be less important information about the document .There are many ways to feature selection. Removing stop words as mentioned before is one way to eliminate unimportant features [1]. Computing term-goodness based on the statistical characteristics of the dataset such as document frequency, information gain, and mutual information is another way [10]. A threshold method, as a method of feature selection is based on removing some features, the removal will be based on the frequencies of those features by setting that frequencies be greater than or less than a defined threshold value. Examples of threshold methods are: document frequency thresholding and chi-square.

In information theory methods, the least predictable terms carry the greatest information value. The least predictable terms are those that exist with the smallest probabilities. Information theory concepts have been used to derive a measure called signal-noise ratio, of term usefulness for indexing (need re-phrasing) [16].

2) Heuristic based selection techniques. Other feature selection techniques uses heuristic information to calculate the similarity and relations that can exist between the features in a text document , stemming techniques that extracts the word's roots, and domain ontology that is based on semantic relations between the features are two examples of heuristic techniques. There are indicators to the importance of features in a document such as term frequency (TF), inverse document frequency (IDF), and their multiplicative combination (TF×IDF) [1].

In the linguistic approach, it simulates the behavior of a linguist by considering Arabic morphological system and analyzing Arabic words according to their morphological components. In this approach, prefix and suffix of a given word are removed by comparing the leading and trailing characters with a given list of affixes in table.

D. Stemming

Stemming is any process to strip additives from the word, In English and English like languages stemming is the process of stripping suffixes from word, however Arabic language words may have additives anywhere in the word and not only suffixes which complicates the stemming task, to ease the process of stemming many researchers introduced light stemming for Arabic language which concentrated on removing all or subset of the affixes (prefixes and suffixes) without touching the additives in the middle of the word (infixes).

Statistical stemmers did not work well for Arabic language while for English and English like languages achieved great results. On the other hand, morphological approaches generate the Arabic word root or set of possible roots. Recently Shawakfa et al. [12] conducted a research that compare different approaches of root finding but most of these approaches generate incorrect root. In the combinational approach, the word to be stemmed is used to generate all possible combinations of letters. Those combinations are matched against predefined lists of Arabic roots. If there is a match, stem and patterns are extracted [18].

Arabic stemming algorithms can be classified as: stem-based, root-based algorithms. Stem–base algorithms basically work by removing all prefixes and suffixes from Arabic words, while on the other hand the root-based algorithms work by reducing stems to roots. Light stemming is the process of stripping off a small set of prefixes and/or suffixes without trying to deal with infixes or recognize patterns and find roots.

Stemming reduces the number of features in a document. Stemming is a computational process that collects all words which share the same stem and have the some semantic relation [14]. The goal of the stemming process is to remove all possible affixes, so as a result reducing the word to its stem. Stemming is usually used for document matching and categorization by finding the standard form of a word in a document and select as a representative for all words of that standard form. There exist many stemming techniques: table lookup, linguistic, and combinational techniques. In table lookup approach, there is a list which consists of all valid Arabic words along with their morphological decompositions. Simply, for a given word it accesses the list and retrieves the associated root/stem. In this case the resulted stem is guaranteed to be accurate. But the backward with this technique is that it is not possible to build a table that has all language words.

V. USED CLASSIFYING METHODOLOGY

The goal of document categorization is to assign documents to a pre-defined and fixed set of documents [1]. Document categorization involves the process of automatically learning categorization patterns so that the categorizations of new documents will be trivial. Categorization models can be divided into three types: the first type identified by "older models" which consists of Boolean and vector space models. The second type is identified by "probabilistic models" which consists of BM25 and language models. The third type is

identified by "combining evidence models" which consists of inference networks and learning to rank models [20].

Nearest Neighbor learners are considered to be lazy learners as they delay the process of modeling the training data until a new document is classified. Rote classifier is an example of a lazy learner, which memorizes the entire training data and does classification only if the features (attributes) of a test document match one of the training documents exactly.

Nearest-neighbor classification technique is part of the instance-based learning technique, which basically uses training documents to make predictions for tested documents without having a model derived from data. Instance-based learning techniques require a proximity measure to determine the similarity between the training documents and the classification function which returns the predicted class of the document under testing based on its proximity to other training documents [15].

KNN classifier is chosen to implement the system for the following reasons: it's simple, similarity measure is reasonable, and doesn't need any resources for training despite some disadvantages such as the above-average categorization time because there was no time invested in the learning phase [1].

The focus of this study is on Vector Space Model (VSM). In VSM, both training documents and tested documents are represented as vectors. Each term in a document is given a weight; the weight indicates the importance of the term in both the document and within the documents in the whole collection of documents.

In this context, q refers to a tested document. A document D_i in the collection of documents and a tested document q can both be represented as vectors, $D_i = (d_{i1}, d_{i2}, \ldots, d_{it})$ and $q = (q_1, q_2, \ldots, q_t)$, where t is the number of index terms in the collection, each d_{ij} and q_j represents document term and tested document term weights respectively.

A. Term Weighting

There are many approaches for term weighting. In this work, a well-known approach called *tf*idf* is used, which is given by equation (1) [21].

$$w_{i,j} = tf_{i,j} * idf_j = tf_{i,j} * \log_{10}(N/df_j) \quad (1)$$

Where $w_{i,j}$ is the weight of term j in document i, $tf_{i,j}$ is the number of times a term j occurs in a document i, idf_j is the number of documents in which the term j appears, and N is the total number of documents in the collection.

Since documents in the collection of text documents does not have the same length (i.e., number of features in documents are not the same), short documents might not have the same chance to be recognized as relevant as long documents; because of this, the retrieval of any document must be made independent of its length; this can be done by normalizing document vectors. So, this makes it fair to retrieve documents of all different lengths. The $tf_{i,j}$ (the raw frequency) is normalized by dividing the raw frequency of the term by the raw frequency of the most common term in the document ($tf_{i,j}/\max(tf_{i,j})$). So, the new term weight is represented by equation (2).

$$w_{i,j} = (tf_{i,j}/\max(tf_{i,j})) * \log_{10}(N/df_j) \quad (2)$$

This way, terms' weights are restricted to be between zero and one; higher weight approach one indicates that the term is important whereas weight approaches zero indicates less important term [22].

B. Similarity Measures

Once the weights for terms in all documents in the collection of text documents are computed, a ranking function is needed to measure the similarity between training document vectors and tested documents. There exist many ranking functions such as Cosine similarity, Euclidean distance, Dice coefficient, Jaccard measure, and Manhattan distance. In this work, cosine measure is used [21]. Cosine measure is one of the most frequently used similarity measures; it calculates the cosine of the angle between the vector of the document and the vector of a tested document. The cosine measure is computed by equation (3).

$$\text{Cos-Sim}(D_i, q) = \frac{\overrightarrow{D_i} \bullet \vec{q}}{\left|\overrightarrow{D_i}\right| \times \left|\vec{q}\right|} = \frac{\sum_{j=1}^{t} d_{ij} \times q_j}{\sqrt{\sum_{j=1}^{t} d_{ij}^{\,2}} \times \sqrt{\sum_{j=1}^{t} q_j^{\,2}}} \quad (3)$$

Where vector D_i represent a training document Di and vector q represents a tested document q. After similarity calculation, documents are then ranked by decreasing cosine value.

C. Evaluation Measurements

The effectiveness of Text Categorization techniques is measured using IR evaluation metrics, such as Recall, Precision, Fallout, Error rate and F measure [11]. Recall is defined as the percentage of relevant documents retrieved out of all the relevant documents in the collection whereas precision is defined as the percentage of relevant documents retrieved out of all retrieved documents. The F-measure is the harmonic mean of the recall and precision.

Assume the case of a binary classification problem where there is only one category and n documents to be classified, then any of the n documents might or might not belong to that category; a document is considered a positive example if it belongs to that category and a negative example in case that document does not belong to that category. So, the documents that have already been classified (given a category) were classified by human experts (Human classifier), beside a computer program will categorize those categorized documents. So, the comparison between human classifier and the program classifier is done by means of recall, precision, fallout, error rate, and F-measure. Those measures are shown in Table IV [5].

TABLE IV. COMMON PERFORMANCE MEASURES

Performance measure	Definition
Recall	$R_i = a_i / (a_i + c_i)$
Precision	$P_i = a_i / (a_i + b_i)$
Fallout	$F_i = b_i / (b_i + d_i)$
Error rate	$Err_i = (b_i + c_i) / (a_i + b_i + c_i + d_i)$
F-measure	$Fm_i = (2 * R_i * P_i) / (R_i + P_i)$

Where a_i is the number of documents correctly assigned to category *i*, b_i is the number of documents incorrectly assigned to category *i*, c_i is the number of documents correctly rejected from category *i*, and d_i is the number of documents incorrectly rejected from category *i*.

D. Dataset Used

The proposed approach is tested using 1000 normalized documents collected from different digital Arabic newspapers. The 1000 documents are equally distributed over five categories: Arts, Politics, Science, Economics, and Sports. In this work three types of word indexing are used: full-word, root, and stem; the stem is obtained by removing prefixes and suffixes from Arabic words (features). In this work, the stemmer proposed by [19] is used. Table V shows the statistics of the Arabic text collection.

TABLE V. ARABIC TEXT COLLECTION STATISTICS

Number of documents	1,000		
Number of words in the collection (excluding stop-words)	204,818		
Type of index-term	***Full-word***	***Stem***	***Root***
Collection size	1.35 MB	1 MB	990 KB
Number of distinct words in the collection (excluding stop-words)	39,819	12,502	13,113

The proposed system is tested for each indexing type using 10-fold cross validation. In every fold, the same number of documents from each category is chosen as tested documents and the remaining are used as training documents, so each document will have the chance to be included in the test collection.

E. Cross-Validation

In this approach, a document is used the same number of times for training and just once for testing. Here the documents are divided into two subsets: one subset for training and the other for testing. Then the role of the two subsets is swapped so that the previous test subset becomes training and the other training subset becomes testing subset. In this work, the corpus is partitioned to be 9/10 as training subset and 1/10 as test subset. Also k-fold cross-validation method is used in which during each run, one of the partitions is selected for testing. While the rest of the documents used for training. This approach is repeated k times so that each partition is used for testing exactly once. In this work, 10-fold cross-validation is used.

F. Classifier Construction

The following are the steps used to build the classifier:

a) Building the index of all documents in the collection. This step involves grouping terms in each document by finding the count of each term in each document.

b) Find the number of documents where each term occur.

c) Find the weight of each term according to the following formula:

d) $w_{ij} = f_{ij} * \lg(N/n_i)$ where f_{ij} is the number of times a term occur in a document, n_i is the number of documents a term occur, and N is the number of documents in the collection.

e) Join the training documents with tested documents based on common terms.

f) Build cosine similarity measure using equation:

The nearest K neighbors among all training documents are determined as a result of calculation. Those K neighbors may be of different categories so the document will be assigned to the category that has the maximum number of documents included in the K nearest neighbors. The similarity measure used in this work is Cosine similarity measure and the value of K used is 80.

VI. EXPERIMENTS AND RESULTS

Table VI shows recall, precision, fallout, and error rate over 10-folds for each category. Also the table shows that recall reaches its highest (0.98) for art category, and the lowest value (0.85) for the politics category. On the other hand, precision reaches its highest for sport (0.99), and the lowest is (0.87) for art. Table VII shows recall, precision, fallout, and error rate over 10-folds for each category. Also the table shows that recall reaches its highest (0.99) for sport category, and the lowest value (0.86) for the politics category. On the other hand, precision reaches its highest for sport (1), and the lowest is (0.90) for economics. Table VIII shows recall, precision, fallout, and error rate over 10-folds for each category. Also the table shows that recall reaches its highest (0.98) for sport category, and the lowest value (0.88) for the politics category. On the other hand, precision reaches its highest for sport (0.99), and the lowest is (0.92) for economics.

After looking at Tables VI, VII, and VIII, one can conclude that politics showed to have minimum recall for full-word, root, and stem indexing whereas sport showed maximum precision for full-word, root, and stem indexing.

Table IX shows the min, max, average for 10-folds for each one of the five categories where 9/10–1/10 ratio used for training/test ratio and using full-word term indexing. Table X shows the min, max, average for 10-folds for each one of the five categories where 9/10–1/10 ratio used for training/test ratio and using root term indexing. Table XI shows the min, max, average for 10-folds for each one of the five categories where 9/10–1/10 ratio used for training/test ratio and using stem term indexing.

Figures 2, 3, 4, 5, 6, and 7 show the usefulness of using cross-validation where document will have the chance to be chosen to a tested document. The figures show the variations of averages of recall and precision for each category spanning over 10-folds using full-word, stem, and root indexing.

TABLE VI. RECALL, PRECISION, FALLOUT, AND ERROR RATE OVER 10-FOLDS FOR EACH CATEGORY USING FULL-WORD INDEXING

Category	**Recall**	**Precision**	**Fall Out**	**Error Rate**
Art	0.9892	0.8750	0.0025	0.0270
Economic	0.9496	0.9400	0.0125	0.0220
Politics	0.8587	0.9450	0.0413	0.0440
Science	0.9729	0.9700	0.0075	0.0120
Sport	0.9772	0.9900	0.0063	0.0070

TABLE VII. RECALL, PRECISION, FALLOUT, AND ERROR RATE OVER 10-FOLDS FOR EACH CATEGORY USING ROOT INDEXING

Category	Recall	Precision	Fall Out	Error Rate
Art	0.9657	0.9150	0.0088	0.0240
Economic	0.9565	0.9050	0.0100	0.0270
Politics	0.8621	0.9300	0.0413	0.0470
Science	0.9729	0.9750	0.0075	0.0110
Sport	0.9952	1.0000	0.0012	0.0010

TABLE VIII. RECALL, PRECISION, FALLOUT, AND ERROR RATE OVER 10-FOLDS FOR EACH CATEGORY USING STEM INDEXING

Category	Recall	Precision	Fall Out	Error Rate
Art	0.9712	0.9300	0.0075	0.0200
Economic	0.9398	0.9250	0.0150	0.0270
Politics	0.8823	0.9269	0.0325	0.0409
Science	0.9577	0.9700	0.0113	0.0150
Sport	0.9861	0.9900	0.0038	0.0050

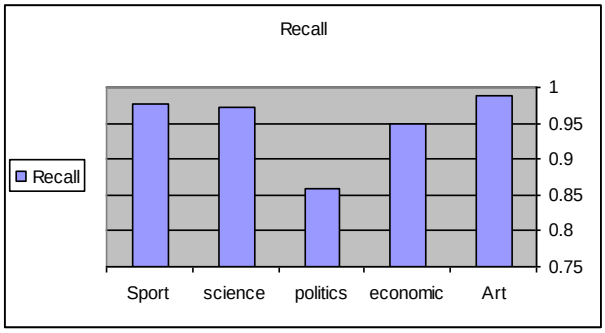

Fig. 2. Variations of average of recall over 10-folds for each category using full-word indexing

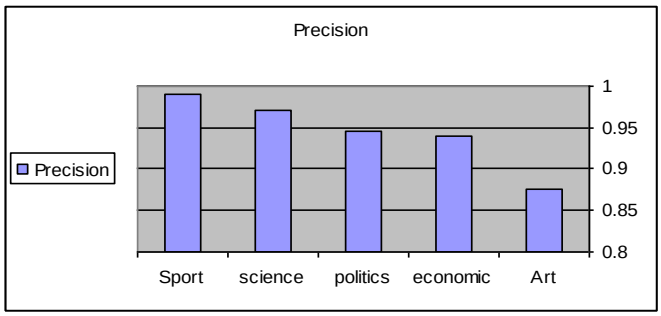

Fig. 3. Variations of average of precision over 10-folds for each category using full-word indexing

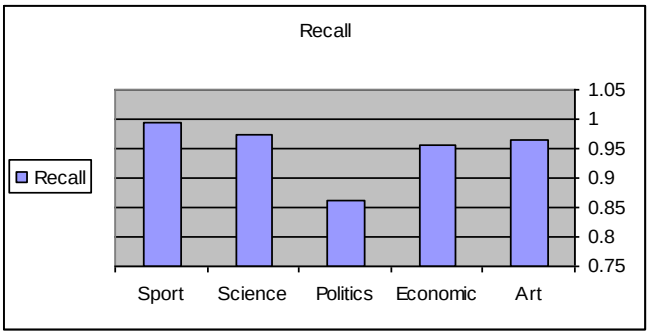

Fig. 4. Variations of average of recall over 10-folds for each category using root indexing

TABLE IX. MAX, MIN, AND AVERAGE FOR 10-FOLD USING FULL-WORD FOR ALL CATEGORIES

Category	Iteration No.	Recall	Precision	Category	Iteration No.	Recall	Precision
Art	1	1.0000	0.7500	Economic	1	0.9474	0.9000
Art	2	1.0000	0.9500	Economic	2	0.9524	1.0000
Art	3	0.9444	0.8500	Economic	3	0.9444	0.8500
Art	4	1.0000	0.7500	Economic	4	0.9524	1.0000
Art	5	1.0000	0.9000	Economic	5	1.0000	1.0000
Art	6	1.0000	0.9000	Economic	6	1.0000	1.0000
Art	7	1.0000	0.8500	Economic	7	1.0000	1.0000
Art	8	1.0000	0.9000	Economic	8	0.8889	0.8000
Art	9	0.9474	0.9000	Economic	9	0.8636	0.9500
Art	10	1.0000	1.0000	Economic	10	0.9474	0.9000
Maximum		***1.0000***	***1.0000***	***Maximum***		***1.0000***	***1.0000***
Minimum		***0.9444***	***0.7500***	***Minimum***		***0.8636***	***0.8000***
Average		***0.9892***	***0.8750***	***Average***		***0.9496***	***0.9400***
Politics	1	0.7692	1.0000	Science	1	1.0000	1.0000
Politics	2	0.9500	0.9500	Science	2	1.0000	0.8500
Politics	3	0.7600	0.9500	Science	3	1.0000	1.0000
Politics	4	0.7917	0.9500	Science	4	1.0000	1.0000
Politics	5	0.9091	1.0000	Science	5	1.0000	1.0000
Politics	6	0.9524	1.0000	Science	6	1.0000	1.0000
Politics	7	0.9000	0.9000	Science	7	0.8696	1.0000
Politics	8	0.7500	0.9000	Science	8	0.9500	0.9500
Politics	9	0.8571	0.9000	Science	9	1.0000	0.9000
Politics	10	0.9474	0.9000	Science	10	0.9091	1.0000
Maximum		***0.9524***	***1.0000***	***Maximum***		***1.0000***	***1.0000***
Minimum		***0.7500***	***0.9000***	***Minimum***		***0.8696***	***0.8500***
Average		***0.8587***	***0.9450***	***Average***		***0.9729***	***0.9700***
Sport	1	1.0000	1.0000				
Sport	2	0.8696	1.0000				

Sport	3	1.0000	0.9500				
Sport	4	1.0000	1.0000				
Sport	5	1.0000	1.0000				
Sport	6	0.9524	1.0000				
Sport	7	1.0000	1.0000				
Sport	8	1.0000	1.0000				
Sport	9	0.9500	0.9500				
Sport	10	1.0000	1.0000				
Maximum		***1.0000***	***1.0000***				
Minimum		***0.8696***	***0.9500***				
Average		***0.9772***	***0.9900***				

TABLE X. MAX, MIN, AND AVERAGE FOR 10-FOLD USING ROOT FOR ALL CATEGORIES

Category	**Iteration No.**	**Recall**	**Precision**	**Category**	**Iteration No.**	**Recall**	**Precision**
Art	1	1	0.7	Economic	1	0.882353	0.75
Art	2	1	0.95	Economic	2	0.95	0.95
Art	3	1	0.95	Economic	3	0.947368	0.9
Art	4	0.947368	0.9	Economic	4	0.952381	1
Art	5	0.95	0.95	Economic	5	1	1
Art	6	0.857143	0.9	Economic	6	1	1
Art	7	1	0.85	Economic	7	1	1
Art	8	0.95	0.95	Economic	8	0.928571	0.65
Art	9	0.952381	1	Economic	9	0.904762	0.95
Art	10	1	1	Economic	10	1	0.85
Maximum		***1***	***1***	***Maximum***		***1***	***1***
Minimum		***0.857143***	***0.7***	***Minimum***		***0.882353***	***0.65***
Average		***0.965689***	***0.915***	***Average***		***0.956544***	***0.905***
Politics	1	0.655172	0.95	Science	1	1	1
Politics	2	0.904762	0.95	Science	2	1	0.95
Politics	3	0.869565	1	Science	3	1	0.95
Politics	4	0.9	0.9	Science	4	1	1
Politics	5	0.95	0.95	Science	5	1	1
Politics	6	0.894737	0.85	Science	6	1	1
Politics	7	0.9	0.9	Science	7	0.869565	1
Politics	8	0.692308	0.9	Science	8	0.95	0.95
Politics	9	0.95	0.95	Science	9	1	0.9
Politics	10	0.904762	0.95	Science	10	0.909091	1
Maximum		***0.95***	***1***	***Maximum***		***1***	***1***
Minimum		***0.655172***	***0.85***	***Minimum***		***0.869565***	***0.9***
Average		***0.862131***	***0.93***	***Average***		***0.972866***	***0.975***
Sport	1	1	1				
Sport	2	0.952381	1				
Sport	3	1	1				
Sport	4	1	1				
Sport	5	1	1				
Sport	6	1	1				
Sport	7	1	1				
Sport	8	1	1				
Sport	9	1	1				
Sport	10	1	1				
Maximum		***1***	***1***				
Minimum		***0.952381***	***1***				
Average		***0.995238***	***1***				

TABLE XI. MAX, MIN, AND AVERAGE FOR 10-FOLD USING STEM FOR ALL CATEGORIES

Category	**Iteration No.**	**Recall**	**Precision**	**Category**	**Iteration No.**	**Recall**	**Precision**
Art	1	1	0.8	Economic	1	0.9	0.9
Art	2	1	0.95	Economic	2	0.95	0.95
Art	3	0.95	0.95	Economic	3	0.9	0.9
Art	4	1	0.85	Economic	4	0.95	0.95
Art	5	0.904762	0.95	Economic	5	1	0.95
Art	6	0.904762	0.95	Economic	6	0.952381	1
Art	7	1	0.95	Economic	7	1	1

Art	8	1	0.9	Economic	8	0.941176	0.8
Art	9	0.952381	1	Economic	9	0.857143	0.9
Art	10	1	1	Economic	10	0.947368	0.9
Maximum		***1***	***1***	***Maximum***		***1***	***1***
Minimum		***0.904762***	***0.8***	***Minimum***		***0.857143***	***0.8***
Average		***0.97119***	***0.93***	***Average***		***0.939807***	***0.925***
Politics	1	0.772727	0.894737	Science	1	0.909091	1
Politics	2	0.904762	0.95	Science	2	1	0.9
Politics	3	0.863636	0.904762	Science	3	1	0.95
Politics	4	0.826087	0.95	Science	4	1	1
Politics	5	0.9	0.947368	Science	5	0.952381	1
Politics	6	0.944444	1	Science	6	1	1
Politics	7	0.947368	1	Science	7	0.909091	1
Politics	8	0.72	0.818182	Science	8	0.95	0.95
Politics	9	0.944444	0.894737	Science	9	0.947368	0.9
Politics	10	1	0.909091	Science	10	0.909091	1
Maximum		***1***	***1***	***Maximum***		***1***	***1***
Minimum		***0.72***	***0.818182***	***Minimum***		***0.909091***	***0.9***
Average		***0.882347***	***0.926888***	***Average***		***0.957702***	***0.97***
Sport	1	1	1				
Sport	2	0.909091	1				
Sport	3	1	0.95				
Sport	4	1	1				
Sport	5	1	0.95				
Sport	6	1	1				
Sport	7	1	1				
Sport	8	1	1				
Sport	9	0.952381	1				
Sport	10	1	1				
Maximum		***1***	***1***				
Minimum		***0.909091***	***0.95***				
Average		***0.986147***	***0.99***				

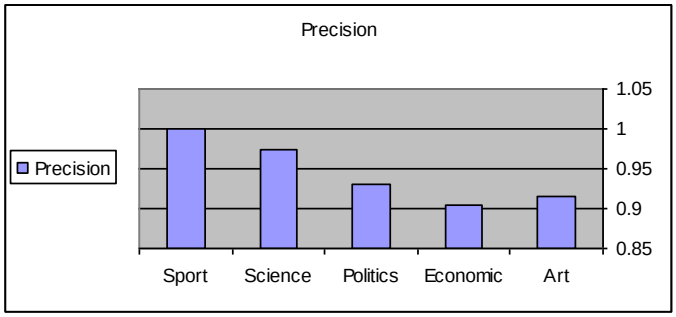

Fig. 5. Variations of average of precision over 10-folds for each category using root indexing

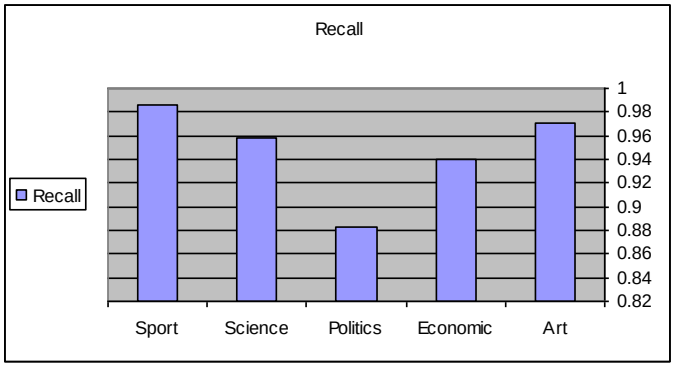

Fig. 6. Variations of average of recall over 10-folds for each category using stem indexing

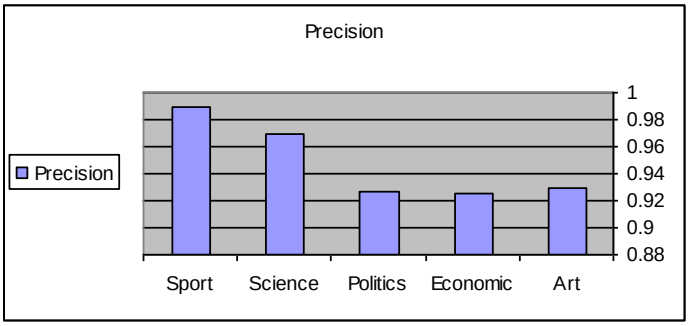

Fig. 7. Variations of average of precision over 10-folds for each category using stem indexing

VII. Conclusion and Future Work

This paper has presented a KNN classifier for Arabic text categorization. Since Arabic language has rich morphology, processing Arabic text was not a trivial task; as we have seen that a single word can have many formations and also a letter in a word can have many styles depends on the location where the letter occur in a word. The classifier is tested against 1000 documents consists of five categories. Vector space model is used to model data. Stop word removal and document frequency threshold methods used for feature selection and reduction. Full-word, stem, and root used for term indexing. In future, we are looking for using the concept of ontology's for enhancing the classifier performance.

REFERENCES

[1] H. Brucher, G. Knolmayer, and M. A. Mittermayer, "Document classification methods for organizing explicit knowledge," in Organizational Knowledge, Learning, and Capabilities, Athens, Greece, 2002, pp. 1-26.

[2] F. Sebastiani , "Machine learning in automated text categorization," ACM Computing Surveys, vol. 34(1), 2002.

[3] L. Khreisat, "A machine learning approach for Arabic text classification using N-gram frequency statistics," Journal of Informetrics, vol. 3(1), pp. 72-77, 2009.

[4] R. Duwairi, "Arabic Text Categorization," The International Arab Journal of Information Technology, vol. 4(2), 2007.

[5] A. Al-Badarneh, E. Al-Shawaka, B. Bani-Ismail, K. Al-Rababah, and S. Shatnawi, "The impact of indexing approaches on Arabic text classification," Journal of Information Science, in press.

[6] G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer, "An KNN Model-based Approach and Its Application in Text Categorization," in Computational Linguistics and Intelligent Text Processing, 2004, pp. 559-570.

[7] R. Al-Shalabi and R. Obeidat, "Improving KNN Arabic Text Classification with N-Grams Based Document Indexing," in Informatics and Systems, 2008, pp. 108-112.

[8] M. Syiam, Z. Fayed, and M. Habib, "An intelligent system for Arabic text categorization," Int. Journal of Intelligent Computing and Information Sciences, vol. 6(1), pp. 1-19, 2006.

[9] I. Hmeidi, B. Hawashin, and E. El-Qawasmeh, "Performance of KNN and SVM classifiers on full word Arabic articles", Advanced Engineering Informatics, vol. 22(1), pp. 106-111, 2008.

[10] R. Duwairi, M. Al-Refai, and N. Khasamneh, "Stemming Versus Light Stemming as Feature Selection Techniques for Arabic Text Categorization," in Innovations in Information Technology, 2007, pp. 446-450.

[11] R. Duwairi, "Machine learning for Arabic Text Categorization," Journal of the American society for information science and technology, vol. 57(8), pp. 1005-1010, 2006.

[12] E. Al-Shawaka, A. Al-Badarneh, S. Shatnawi, K. Al-Rababah, and B. Bani-Ismail, "A comparison Study of Some Arabic Root Finding Algorithms", Journal of the American society for information science and technology, vol. 61(5), pp. 1005-1010, 2010.

[13] N. Abouzakhar, B. Allison, and L. Guthrie "Unsupervised Learning-based Anomalous Arabic Text Detection," in Language Resources and Evaluation, 2008, pp. 291-296.

[14] R. Al-Shalabi et al., "Proper Noun Extracting Algorithm for Arabic Language," in IT to Celebrate S. Charmonman's 72nd Birthday, 2009.

[15] P. Tan, M. Steinbach, and V. Kumar, Introduction to Data Mining, Addison-Wesley, USA, 2005.

[16] K. Fuka and R. Hanka, "Feature set reduction for document classification problems," in Artificial Intelligence, 2001, pp. 1-7.

[17] H. Moukdad, "Stemming and root-based approaches to the retrieval of Arabic documents on the Web," Webology, vol. 3(1), 2006.

[18] H. Al-Sherhan and A. Ayesh, "A triliteral Word Roots Extraction Using Neural Network for Arabic," in Computer Engineering and Systems, 2006, pp. 436-440.

[19] S. Ghwanmeh, G. Kanaan, R. Al-Shalabi, and S. Rabab'ah, "Enhanced algorithm for extracting the root of Arabic words," in Computer Graphics, Imaging and Visualization: New Advances and Trends, 2009, pp. 388-391.

[20] B. Croft, D. Metzler, and T. Strohman, Search engines: information retrieval in practice, Addison-Wesley, USA, 2009.

[21] D. Lee, H. Chuang, and K. Seamons, "Document ranking and the vector-space model," IEEE Software, vol. 14(2), pp. 67-75, 1997.

[22] F. Harrag, A. Hamdi-Cherif, and E. El-Qawasmeh, "Vector space model for Arabic information retrieval–application to hadith indexing," in the Applications of Digital Information and Web Technologies, 2008, pp. 107-112.