An Intelligent System For Automated Arabic Text Categorization

An Intelligent System For Arabic Text Categorization

Article · January 2006

Source: OAI

CITATIONS
124

READS
323

3 authors, including:

Mena Badieh Habib Morgan
Maastricht University
30 PUBLICATIONS **363** CITATIONS

SEE PROFILE

IJICIS, Vol.6, No. 1, JANUARY 2006

AN INTELLIGENT SYSTEM FOR ARABIC TEXT CATEGORIZATION

M. M. Syiam, Z. T. Fayed & M. B. Habib

Computer Science Department
Faculty of Computer and Information Sciences
Ain Shams university,
Cairo, Egypt
menabad@yalla.com

Abstract: *Text Categorization (classification) is the process of classifying documents into a predefined set of categories based on their content. In this paper, an intelligent Arabic text categorization system is presented. Machine learning algorithms are used in this system. Many algorithms for stemming and feature selection are tried. Moreover, the document is represented using several term weighting schemes and finally the k-nearest neighbor and Rocchio classifiers are used for classification process. Experiments are performed over self collected data corpus and the results show that the suggested hybrid method of statistical and light stemmers is the most suitable stemming algorithm for Arabic language. The results also show that a hybrid approach of document frequency and information gain is the preferable feature selection criterion and normalized-tfidf is the best weighting scheme. Finally, Rocchio classifier has the advantage over k-nearest neighbor classifier in the classification process. The experimental results illustrate that the proposed model is an efficient method and gives generalization accuracy of about 98%.*

Keywords *Text mining, text categorization, machine learning, stemming, feature selection.*

1. INTRODUCTION

Text Categorization (classification) is the process of classifying documents into a predefined set of categories based on their content. This assignment can be used for classification, filtering, and retrieval purposes. Machine learning approaches are applied to build an automatic text classifier by learning from a set of previously classified documents [1]. Many text categorization systems have been developed for English and other European languages, but according to a performed survey there are few researches for Arabic text categorization till the day of writing this paper. The document in text categorization system must pass through a set of steps: document conversion which converts different types of documents into plain text, stop word removal to remove insignificant words, stemming to group words sharing the same root, feature selection/extraction, super vector construction, feature weighting, classifier construction, classification, evaluation of the classifier.

Arabic language is a Semitic language that has a complex and much morphology than English, it is a highly inflected language, and due to this complex morphology it needs a set of preprocessing routines to be suitable for manipulation. Stop words like prepositions and particles are considered insignificant words and must be removed, words must be stemmed after stop words removal. Stemming is the process of removing the affixes from the word and extracting the word root [2,3,4,5,6,7,8]. After applying preprocessing

routines, document is passed by document indexing process which involves creation of internal representation of the document. Indexing process consists of three phases [9]:

(a) Construction of the super vector which is the vector containing all terms that appears in all the documents in the corpus;
(b) Term selection which can be seen as a form of dimensionality reduction by selecting a subset of terms from the full original set of terms in the super vector according to some criteria, this subset are expected to yield the best effectiveness, or the best compromise between effectiveness and efficiency [10,11,12];
(c) Term weighting in which, for every term selected in phase (b) and for every document, a weight is computed which represents how much this term contributes to the discriminative semantics of the document [13].

Finally, the classifier is constructed by learning the characteristics of every category from a training set of documents. Once a classifier has been built, its effectiveness (i.e. its capability to take the right categorization decisions) may be tested by applying it to the test set and checking the degree of correspondence between the decisions of the classifier and those encoded in the corpus.

S1`a`

The paper is organized as follow. Section 2 briefly shows the nature and some morphological samples in the Arabic language. A general review on related work in text categorization is presented in section 3. Section 4 presents the proposed model for Arabic text categorization. The achieved experimental results are discussed in section 5. Finally, conclusion and future work are presented in section 6.

2. ARABIC LANGUAGE STRUCTURE

Arabic is the mother language of more than 300 million people [2]. Unlike Latin-based alphabets, the orientation of writing in Arabic is from right to left, the Arabic alphabet consists of 28 letter. Arabic language is a highly inflected language, it has much richer morphology than English. Arabic words have two genders, feminine and masculine; three numbers, singular, dual, and plural; and three grammatical cases, nominative, accusative, and genitive. 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.

Words are classified into three main parts of speech, nouns (including adjectives and adverbs), verbs, and particles. All verbs and some nouns are morphologically derived from list of roots. Words are formed by following fixed patterns, the prefixes and suffixes are added to the word to indicate its number, gender and tense.

Most of Arabic words are derived from the pattern *Fa'ala* (فعل), all words following the same pattern have common properties and states. For example the pattern *Faa'el* (فاعل) indicates the subject of the verb, the pattern *maf'ool* (مفعول) represents the object of the verb. Table 1 shows 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 these derivations on the meaning. The letters that have been added to the main root of the word are underlined.

Table 1: Different derivations for the root word *kataba* (كتب)

Arabic word	Pattern	Pronunciation	English meaning
كتب	*Fa'ala* (فعل)	*Kataba*	Wrote
كتابة	*Fe'ala*	Ketaba	Writing
كاتب	*Fa'el* (فاعل)	Kateb	Writer
مكتوب	*Maf'ool* (مفعول)	*Maktoob*	Is written
كتاب	*F'aal* (فعال)	*Ktaab*	Book
مكتبة	*Maf'ala* (مفعلة)	*Maktaba*	Library
مكتب	*Maf'al* (مفعل)	*Maktab*	Office

In addition to the different forms of the Arabic word that result from the derivational process, most connectors, conjunctions, prepositions, pronouns, and possession forms are attached to the Arabic surface form as prefixes and suffixes. For instance, the definitive nouns are formed by attaching the article (ال) to the immediate front of the nouns (act as "The"). The conjunction word (و) (and) is often attached to the word. The letters (ف ، ل ، ب ، ك) can be added to the front to the word as prepositions. The suffix (ة) is attached to represent the feminine of the word, (ان) is for dual masculine in the nominative case, (ين) is for dual masculine in both the accusative and the genitive cases, (ون) is for plural masculine in the nominative case, and (ين) for plural masculine in the accusative or genitive cases. The plural suffix (ات) is used in case of feminine gender for the three grammatical cases. Also some suffixes are added as possessive pronouns, the letter (هـ) is added to represent the possessive pronoun (His), (ها) for (Her), (ي) for (My), and (هم، هن) for (Their). Table 2 shows different affixes that may be added to the word (معلم) (Teacher), the affixes attached to the word are underlined, also the table shows the corresponding meaning of the word in English along with its gender and number state.

Table 2: Different affixes that may be added to the word (معلم).

Arabic word	English meaning	Gender	Number
معلم	Teacher	Masculine	Singular
معلمة	Teacher	Feminine	Singular
معلمان	Two teachers	Masculine	Dual
معلمون	Teachers	Masculine	Plural (accusative, genitive)
معلمين	Teachers	Masculine	Plural (nominative)
معلمات	Teachers	Feminine	Plural
المعلم	The teacher	Masculine	Singular
والمعلم	And the teacher	Masculine	Singular
كالمعلم	Like the teacher	Masculine	Singular
معلمي	My teacher	Masculine	Singular
معلمه	His teacher	Masculine	Singular
معلمها	Her teacher	Masculine	Singular
معلمهم	Their teacher	Masculine	Singular

3. RELATED WORK IN TEXT CATEGORIZATION

Many researchers have been working on text categorization in English and other European languages, however few researchers work on text categorization for Arabic language. Here, a general review on the related work for text categorization and its machine learning techniques is presented.

Stemming is the process of removing all affixes from a word to extract its root. It has shown to improve performance in information retrieval tasks specially with highly inflected language like Arabic Language. Many stemmers that have been developed for English and other European languages, mostly deal with the removal of suffixes as this is sufficient for most information retrieval purposes. Some of the most widely known stemmers for English are Lovins and Porter stemming algorithms. For Arabic language there are three different approaches for stemming: the root-based stemmer; the light stemmer; and the statistical stemmer.

Root-Based stemmer uses morphological analysis to extract the root of a given Arabic word. Many algorithms have been developed for this approach. Al-Fedaghi and Al-Anzi algorithm tries to find the root of the word by matching the word with all possible patterns with all possible affixes attached to it [3]. The algorithm does not remove any prefixes or suffixes. Al-Shalabi morphology system uses different algorithms to find the roots and pattern [4]. This algorithm removes the longest possible prefix, then extracts the root by checking the first five letters of the word. This algorithm is based on an assumption that the root must appear in the first five letters of the word. Khoja has developed an algorithm that removes prefixes and suffixes, all the time checking that it's not removing part of the root and then matches the remaining word against the patterns of the same length to extract the root [5].

The aim of the Light stemming approach is not to produce the root of a given Arabic word, rather is to remove the most frequent suffixes and prefixes. Light stemmer is mentioned by some authors [2,6,7,8], but till now there is almost no standard algorithm for Arabic light stemming, all trials in this field were a set of rules to strip off a small set of suffixes and prefixes, also there is no definite list of these strippable affixes.

In statistical stemmer, related words are grouped based on various string similarities measures. Such approaches often involve n-gram [7,14]. Equivalence classes can be formed from words that share some initial letter n-gram or by refining these classes with clustering techniques. An n-gram is a set of n consecutive characters extracted from a word. The main idea behind this approach is that, similar words will have a high proportion of n-grams in common. Typical values for n are 2 or 3, these corresponding to the use of digrams or trigrams, respectively.

After stop words removal and stemming, documents are indexed. In true information retrieval style, each document is usually represented by a vector of *n* weighted terms, this is often referred to as the bag of words approach to document representation [15]. In this approach the structure of a document and the order of words in the document are ignored. The feature vectors represent the words observed in the documents. The super vector $W (w_1, \ldots, w_d)$ in the training set consists of all the distinct words (also called terms) that

appear in the training samples after removing the stop words and words stemming. Typically, there can be thousands of features in document classification. Hence, a major characteristic, or difficulty of text categorization problems is the high dimensionality of the feature space.
Many term evaluation functions have been introduced for term selection for English text categorization [11,12,13]. These functions are Document Frequency Thresholding, Information Gain, CHI Square, Odds Ratio, NGL Coefficient and GSS Score.

After selecting the significant terms, each term is weighted for every document. Term weighting refers to the different ways to compute term weights. Many weighting schemes are evaluated for English and other languages [16].

For classification, there have been two main approaches to the construction of text categorization systems. First, a number of systems have embodied approaches similar to those used in expert systems for classification or diagnosis. Knowledge engineers define one or more layers of intermediate conclusions between the input evidence (words and other textual features) and the output categories and write rules for mapping from one layer to another, and for confirming or removing conclusions.

The second strategy is to use existing bodies of manually categorized text in constructing categorizers by inductive learning. A wide variety of learning approaches have been used. Learning-based systems have been found to be cheaper and faster to build, as well as more accurate in some applications.

There two different ways to build a classifier:
- Parametric: According to this approach, training data are used to estimate parameters of a probability distribution. The main example of this approach is the probabilistic Naive Bayes classifier.
- Non-parametric: This approach may be further subdivided in two categories:
    - Example-based: According to this approach, the document *d* to be categorized is compared against the training set of documents. The document is assigned to the class of the most similar training documents. Example of this approach is k-NN classifer;

    - Profile-based: In this approach, a profile (or linear classifier) for the category, in the form of a vector of weighted terms, is extracted from the training documents pre-categorized under c_i. The profile is then used as a training data against the documents *D* to be categorized. Example of this approach is Rocchio classifier.

After constructing a classifier, it must be evaluated for the text categorization task. Many different evaluation criteria have been used for evaluating the performance of categorization systems [1,9]. The experimental evaluation of a classifier usually measures its generalization, rather than its efficiency, that is, its ability to take the right classification decisions.

4. THE PROPOSED MODEL FOR ARABIC TEXT CATEGORIZATION

The purpose of this paper is to apply machine learning techniques commonly used with text categorization on Arabic language. The proposed model contains a set of phases that describe the documents

preprocessing routines, document representation techniques and classification process. Figure 1 shows the proposed model for Arabic text categorization.

Documents preprocessing routines include stop word removal to remove insignificant words, stemming to group words share the same root. After that, the super vector is constructed. Feature selection techniques are applied to reduce the dimensionality of the super vector. Document is represented as a vector of weighted terms. Finally, classifier is constructed and evaluated. Every phase will be described in details.

Fig 1**:** The proposed model for Arabic text categorization System.

4. 1. Arabic Stemming Approaches

For stemming, the model provides a comparative study for the Root-Based stemmer, light stemmer and statistical stemmer to decide which approach is suitable for Arabic text categorization task.

4.1.1 Root-based stemmer

All Root-Based stemmers have the same technique which is pattern matching to find the root of the word. The root is extracted after removing the suffixes and the prefixes attached to the given word. The root extraction process is started by matching the positions of the surface word letters that corresponds to a pattern. Figure 2 describes the pattern matching process. After extracting the letters that corresponds to the pattern (*fa'ala*), these letters represents the root.

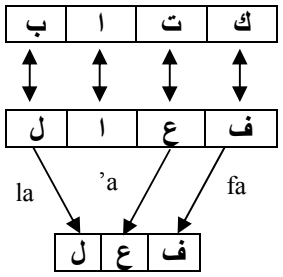

Fig 2: Steps to extract the root letters of the word (كتاب) by pattern matching.

A Root-Based stemming algorithm has been developed. This algorithm removes the most common suffixes and prefixes from the word then matches the word against a set of suggested 67 patterns represent most of word forms. Also the algorithm aims at removing insignificant words from the text, these unvalued words are the stop words, foreign words, and digits. The used Root-Based stemming approach is shown in the following algorithm.

For every word in the text

1. IF the word is not an Arabic word THEN consider this word as a useless word.
2. IF the word contains digits THEN consider this word as a useless word.
3. IF the word length < 3 characters THEN consider this word as a useless word.
4. Remove diacritics.
5. Normalize the word.

6. IF the word is a stop word THEN consider this word as a useless word.
7. Remove prefixes.
8. Recursively remove suffixes.
9. IF the word is a stop word THEN consider this word as a useless word.
10. Match word against 67 patterns and extract the root.

First the algorithm makes sure that the word is an Arabic word, also it considers any word contains less than 3 letters as an article and thus a not important word. Then it removes diacritics (ْ, ِ, ـ, ّ, ُ, ٌ, ً, َ), which are marks above or below letters used in orthography and as a sign for the word grammatical case. After that it begins by normalizing the word as presented by many authors [2,6,7,8]. Normalization is the process of unification of different forms of the same letter as follows.

- Normalize أ , إ , آ to ا .
- Normalize ـة to ـه.
- Normalize ى to ي.
- Normalize the sequence ء ي and the sequence ء ى to ئ.

After word normalization the algorithm checks if the word is one of the stop words list. The stop words list consists of 165 word based mainly on Khoja list [5] plus some words have been added. After stop word elimination, the algorithm removes a set of prefixes (للـ , ال , كال , فال , بال , و) and the letter (ل) if the word starts with the sequence (لا) , after removing these prefixes it checks if the word length is less than 3 letters, in this case this prefix is considered as a main part of the word and so the removed prefix is returned back to the word.

In step 8 the suffixes (هما , كما , ات , يه , ته , تي , ان , ون , ين , هم , هن , ها , نا , وا , كم , كن , ي , ـه) are recursively removed from the tail of the word. The longest suffix is removed first, then the shorter. This process is recursive because most suffixes are compound of pronouns, gender and number suffixes, for example the word (مكتباتهم) (Their libraries) has a composite suffix (اتهم) which is made from two parts (ات) for feminine plural and the pronoun (هم). Also as done in the previous step the algorithm checks if the word length is greater than 3 letters in order not to remove a main part of the word.

After prefixes and suffixes removal the word is checked against the stop words list again because some stop words may have some prefixes and suffixes attached to them. Finally the word is checked against a set of 67 pattern to extract the root.

4.1.2. Light stemmer

Also, a light stemming algorithm is developed. It removes the most common suffixes and prefixes and keeps the form of the word without changing. The same steps used for Root-Based stemming algorithm are used with the light stemmer except the step number 10 in which the root is extracted.

4.1.3. Statistical stemmer

Finally, statistical n-gram stemmer is implemented. The following example shows how digram similarity between the word (سياسة) (Politic) and the word (سياسيا) (Political) is measured.

- سياسة ⇒ سة ، اس ، يا ، سي . (We divided the word into set of digrams each of two adjacent letters)
- Unique digrams ⇒ سة ، اس ، يا ، سي .

- سياسيا ⇒ يا ، سي ، يا ، اس ، سي .
- Unique digrams ⇒ اس ، يا ، سي .

$$\text{Similarity} = \frac{2C}{A+B} = \frac{2*3}{4+3} = 0.8571.$$

Where A and B are the numbers of unique digrams in the first and the second words. C is the number of unique digrams shared by A and B.

Similarity measures are determined for all pairs of terms in the corpus after removing stop and non Arabic words, and applying light stemmer to remove the common affixes. Terms that have a similarity above a predefined threshold are clustered and represented with only one term.

Experimental results shows that the hybrid approach of light stemmer and statistical trigram stemmer (n=3) is the most suitable stemming algorithm for Arabic text categorization system. Reasons for surpass of the hybrid stemmer over the other approaches are discussed in details in the experimental results section.

4.2 Document Indexing

After stop words removal and words stemming, documents are indexed and represented as a vector of weighted terms.

4.2.1 Term selection

Many classifier induction methods are computationally hard, and their computational cost is a function of the length of the vectors that represent the documents. It is thus of key importance to be able to work with vectors with shorter length than the length of the super vector, which is usually a number in the tens of thousands or more. For this, term selection techniques are used to select from the super vector terms a subset of terms that are deemed most useful for compactly representing the meaning of the documents. Usually, these techniques consist in scoring each term in the super vector by means of a term evaluation function *f* (TEF) and then selecting a set of terms that maximize *f*. Often, term selection is also beneficial in that it tends to reduce overfitting, i.e. the phenomenon by which a classifier tends to be better at classifying the data it has been trained on than at classifying other data.
Document Frequency for a word is the number of documents in which the word occurs. In Document Frequency Thresholding one computes the document frequency for each word in the training corpus and removes those words whose document frequency is less than some predetermined threshold that represents how rare is the words (eg. Words with Document Frequency less than 2 are removed). The basic assumption is that rare words are either non-informative for category prediction, or not influential in global performance.

Table 3: Main functions used for term selection purposes.

Function	Denoted by	Mathematical form	
Information Gain	IG (t_k,c_i)	$\sum_{c\in\{c_i,\bar{c_i}\}}\sum_{t\in\{t_k,\bar{t_k}\}} P(t,c).\log\frac{P(t,c)}{P(t).P(c)}$	(4.1)
CHI-square	CHI (t_k,c_i)	$\frac{\lvert T_r\rvert.[P(t_k,c_i).P(\bar{t_k},\bar{c_i})-P(t_k,\bar{c_i}).P(\bar{t_k},c_i)]^2}{P(t_k).P(\bar{t_k}).P(c_i).P(\bar{c_i})}$	(4.2)
NGL coefficient	NGL (t_k,c_i)	$\frac{\sqrt{\lvert T_r\rvert}.[P(t_k,c_i).P(\bar{t_k},\bar{c_i})-P(t_k,\bar{c_i}).P(\bar{t_k},c_i)]^2}{\sqrt{P(t_k).P(\bar{t_k}).P(c_i).P(\bar{c_i})}}$	(4.3)
Odds Ratio	OR (t_k,c_i)	$\frac{P(t_k,c_i).(1-P(t_k,\bar{c_i}))}{(1-P(t_k,c_i)).P(t_k,\bar{c_i})}$	(4.4)
GSS coefficient	GSS (t_k,c_i)	$P(t_k,c_i).P(\bar{t_k},\bar{c_i})-P(t_k,\bar{c_i}).P(\bar{t_k},c_i)$	(4.5)

Other more sophisticated information-theoretic functions have been used in the literature, among them the CHI-square, NGL coefficient, Information Gain, Odds Ratio, and GSS coefficient. The mathematical definitions of these measures are summarized for convenience in Table 3. In these functions, probabilities are interpreted on an event space of documents (e.g., $P(\bar{t_k},c_i)$ denotes the probability that, for a random document x, term t_k does not occur in x and x belongs to category c_i), and are estimated by counting occurrences in the training set. $P(c_i)$ can be estimated from the fraction of documents in the total collection that belongs to class c_j. All functions are specified "locally" to a specific category c_i; in order to assess the value of a term t_k in a "global," category independent sense, either the sum $f_{sum}(t_k) = \sum_{i=1}^{|c|} f(t_k,c_i)$, or the weighted sum $f_{wsum}(t_k) = \sum_{i=1}^{|C|} P(c_i)f(t_k,c_i)$, or the maximum $f_{max}(t_k) = \max_{i=1}^{|C|} f(t_k,c_i)$ of their category-specific values $f(t_k,c_i)$ are usually computed. These functions try to capture the intuition that the best terms for c_i are the ones distributed most differently in the sets of positive and negative examples of c_i. However, interpretations of this principle vary across different functions.

It is found that trying hybrid criteria between DF and other functions offers high accuracy in selecting significant features than using the traditional criteria. Experiments performed shows that a hybrid approach between Document Frequency and Information gain is the best choice for term selection. More details will be presented in the experiments and results section.

4.2.2 Term weighting

Given a collection of documents, its feature vectors are represented by a word-by-document matrix A, where each entry represents the weight of a word in a document, i.e.,

$$A = (a_{ik}) \tag{4.6}$$

Where a_{ik} is the weight of word i in the document k, Since every word does not normally appear in each document, the matrix A is usually sparse. The number of rows, M, of the matrix corresponds to the number of words in the super vector W. M can be very large.

There are several ways of determining the weight a_{ik} of word i in document k, but most of the approaches are based on two empirical observations regarding text [13]:

- The more times a word occurs in a document, the more relevant it is to the topic of the document.
- The more times the word occurs throughout all documents in the collection, the more poorly it discriminates between documents.

Let f_{ik} be the frequency of word i in document k, N the number of documents in the collection, M the number of words in the collection after stop word removal and word stemming, and n_i the total number of times word i occurs in the whole collection. Traditional methods for term weighting are used to determine the most suitable one for Arabic text categorization task.

i. Boolean weighting

The simplest approach is to let the weight be 1 if the word occurs in the document and 0 otherwise:

$$a_{ik} = \begin{cases} 1 & \text{if } f_{ik} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{4.7}$$

ii. Term frequency weighting (tf)

Another simple approach is to use the frequency of the word in the document:

$$a_{ik} = f_{ik} \tag{4.8}$$

iii. Term frequency inverse document frequency weighting (tfidf)

The previous two schemes do not take into account the frequency of the word throughout all documents in the collection. A well-known approach for computing word weights is the tfidf (term frequency-inverse document frequency) weighting which assigns the weight to word i in document k in proportion to the number of occurrences of the word in the document, and in inverse proportion to the number of documents in the collection for which the word occurs at least once.

$$a_{ik} = f_{ik} * \log\left(\frac{N}{n_i}\right) \tag{4.9}$$

iv. Normalized-tfidf weighting

The tfidf weighting does not take into account that documents may be of different lengths. The normalized-tfidf weighting is similar to the tfidf weighting except for the fact that length normalization is used as part of the word weighting formula.

$$a_{ik} = \frac{f_{ik} * \log\left(\frac{N}{n_i}\right)}{\sqrt{\sum_{j=1}^{M}\left[f_{ik} * \log\left(\frac{N}{n_i}\right)\right]^2}} \tag{4.10}$$

By trying all of the above schemas, normalized-tfidf schema is chosen as the best schema for term weighting. More details will be presented in the experiments and results section.

4.3 Text Classifiers

Finally, the proposed model uses two different non-parametric classifiers; k-NN and Rocchio classifiers. Here, these two classifiers are presented , also different classification evaluation criteria will be discussed.

4.3.1 k-Nearest neighbor classifier

To classify an unknown document vector *d*, the k-nearest neighbour (k-NN) algorithm ranks the document's neighbours among the training document vectors, and use the class labels of the k most similar neighbours to predict the class of the input document [1,9]. The classes of these neighbours are weighted using the similarity of each neighbour to *d*, where similarity may be measured by for example the Euclidean distance or the cosine between the two document vectors. The Euclidean distance is used as a conventional method for measuring distance between two documents, the formula of the Euclidean distance between documents $d_1(w_{11},w_{12},...,w_{1n})$ and $d_2(w_{21},w_{22},...,w_{2n})$ is as follow:

$$E(d1,d2) = \sqrt{\sum_{i=1}^{n}(w_{2i} - w_{1i})^2} \tag{4.11}$$

k-NN has been applied to text categorization since the early days of its research. However, it has a set of drawbacks. k-NN is a lazy learning example-based method that does not have a off-line training phase. The main computation is the on-line scoring of training documents given a test document in order to find the k nearest neighbours, this makes k-NN not efficient because nearly all computation takes place at classification time rather than when the training examples are first encountered, k-NN time complexity is O(N*M) where N is number of training documents and M is the number terms for each document vector. Moreover, k-NN classifier has a major drawback of selecting the value of k, the success of classification is very much dependent on this value. The Rocchio method however can deal with those problems to some extent as shown in the next section.

4.3.2 Rocchio classifier

Rocchio is the classic profile-based classifier used for document routing or filtering in information retrieval [17]. In this method, a prototype vector is built for each class c_i, and a document vector *d* is classified by calculating the distance between *d* and each of the prototype vectors [1,9]. The prototype vector for class c_i is computed as the weighted average vector over all training document vectors that belong to class c_i. This means that learning is very fast for this method compared to the k-NN classifier.

The weighted average of a category $c_i(w_{i1},w_{i2},...,w_{in})$ is computed as follow:

$$w_{ik} = \beta \sum_{d_j \in POS_i} \frac{w_{jk}}{|\ POS_i\ |} - \gamma . \sum_{d_j \in NEG_i} \frac{w_{jk}}{|\ NEG_i\ |} \tag{4.12}$$

where w_{jk} is the weight of the term t_k in document d_j, POS_i is the set of documents that belongs to c_i (positive examples), and NEG_i is the set of documents that doesn't belongs to c_i (negative examples). In this formula, β and γ are control parameters that allow setting the relative importance of positive and negative examples. For instance, if β is set to 1 and γ to 0, the profile of c_i is the centroid of its positive training examples. In general, the Rocchio classifier rewards the closeness of a test document to the centroid of the positive training examples, and its distance from the centroid of the negative training examples. The role of negative examples is usually de-emphasized, by setting β to a high value and γ to a low one (e.g. use β=1.6 and γ=0.4) [18].

The Rocchio method deals with k-NN problems to some extent. It uses the generalized instances to replace the whole collection of training instances by summarizing the contribution of the instances belonging to each category. Besides its efficiency this method is easy to implement, since learning a classifier basically comes down to averaging weights and classifying a new instance only needs computing the Euclidean distance between the new instance and the generalized instances. It can be regarded as a similarity-based algorithm. Its time complexity is considered to be O(L*M) where L is number of generalized instances and M is the number terms for each document vector. Moreover, the Rocchio method can deal with noise to some extent via summarizing the contribution of the instances belonging to each category. For example, if a feature mainly appears in many training instances of a category, its corresponding weight in the generalized instance will have a larger magnitude for this category. Also if a feature mainly appears in training instances of other categories, its weight in the generalized instance will tend to zero. Therefore, the Rocchio classifier can distill out certain relevant features to some extent. On the other hand, one drawback of the Rocchio classifier is it restricts the hypothesis space to the set of linear separable hyper-plane regions, which has less expressiveness power than that of k-NN algorithms.

4.3.3 Classifier evaluation

Classification generalization is usually measured in terms of the classic information retrieval notions of precision (π) and recall (ρ) [19], adapted to the case of text categorization. Precision (π_i) with respect to c_i is the probability that if a random document d_x is classified under c_i, this decision is correct. Analogously, recall (ρ_i) with respect to c_i is defined the probability that, if a random document d_x ought to be classified under c_i , this decision is taken. Precision and recall are calculated as follow:

$$\pi_i = \frac{a_i}{a_i + b_i} \qquad \rho_i = \frac{a_i}{a_i + c_i} \tag{4.13}$$

where:
" a " the number of documents correctly assigned to this category.
" b " the number of documents incorrectly assigned to this category.
" c " the number of documents incorrectly rejected from this category.

" d " the number of documents correctly rejected from this category.

For obtaining estimates of π and ρ, two different methods may be adopted:

- Microaveraging: π and ρ are obtained by summing over all individual decisions:

$$\pi^{\mu} = \frac{\sum_{i=1}^{|C|} a_i}{\sum_{i=1}^{|C|} a_i + b_i} \qquad \rho^{\mu} = \frac{\sum_{i=1}^{|C|} a_i}{\sum_{i=1}^{|C|} a_i + c_i} \tag{4.14}$$

- Macroaveraging: π and ρ are first evaluated "locally" for each category, and then "globally" by averaging over the results of the different categories:

$$\pi^{M} = \frac{\sum_{i=1}^{|C|} \pi_i}{|C|} \qquad \rho^{M} = \frac{\sum_{i=1}^{|C|} \rho_i}{|C|} \tag{4.15}$$

Since most classifiers can be arbitrarily tuned to emphasize recall at the expense of precision (and viceversa), only combinations of the two are significant. The most popular way to combine the two is the function $F_{\beta i} = \frac{(B^2+1)\pi_i\rho_i}{\beta^2\pi_i + \rho_i}$, for some value $0 \le \beta \le \infty$; usually, β is taken to be equal to 1, which means that the $F_{\beta i}$ function becomes $F_{1i} = \frac{2\pi_i\rho_i}{\pi_i + \rho_i}$, i.e. the harmonic mean of precision and recall. Similar to precision and recall, F_β function can be estimated using two methods: Microaverage, and Macroaverage

5 EXPERIMENTS AND RESULTS

5.1 Text Collection

We have collected our own text collection. This collection consists of 1,132 documents and contains 39,468 word collected from the three main Egyptian newspapers El Ahram, El Akhbar, and El Gomhoria during the period from August 1998 to September 2004. These documents cover 6 topics. Table 4 shows the number of documents for each topic. Documents have average size of about 117 words before stemming and stop words removal. Document represents the first paragraph of an article, it has been chosen because it usually contains an abstract to the whole article.

Table 4: Number of documents for each topic in the text collection.

Topic	No. of documents
Arts	233
Economics	233
Politics	280
Sports	231
Woman	121
Information Technology	102

5.2 Stemming

Three stemming approaches discussed in section 4.1 have been applied to the text collection to discover which one is suitable for the task of categorizing Arabic documents, also the effect of the stemming on the categorization process is tested versus keeping the words without stemming at all.

Performed experiments uses the different stemming approaches with Document Frequency Thresholding criteria as a default term selection criteria and the boolean weighting for representing the documents and the Rocchio classifier with β=1.6 and γ=0.4 for classifications, the testing method used is the leave one method, the macroaveraged F_1 is used as an evaluation criterion. The macroaveraged F_1 measure is recorded for different number of terms ranged from 2500 to 5000 term selected according the Document Frequency Thresholding criteria.

For statistical n-gram stemming approaches different values for N are used, N=2 (digram) and N=3 (trigram), also different similarities threshold values are used, words with n-gram similarity above that threshold are assumed to be similar and have the same impact in the documents. An improvement has been performed to statistical stemmer by applying light stemmer before performing similarity measure in order to maximize the performance of the statistical stemmer.

Results in figure 3 shows that the hybrid approach of light and trigram stemming with similarity threshold (0.8) is the most suitable stemming approach for Arabic text categorization, results also shows that no stemming, digram stemmer with threshold (0.9) and hybrid stemmer of light stemmer with both digram and trigram stemmers with threshold (0.9) gave poor results that is because those methods didn't mention the similarity of words or that some words may share the same root. On the other hand root stemming gave intermediate accuracy, while light stemming, digram stemmer with thresholds (0.7) and (0,8), and hybrid stemmer of light stemmer with both digram and trigram with similarity thresholds (0.7) and (0.8) gave the best results as it tries to group the words in some how. Light stemming only removes some common prefixes and suffixes, however there are many other rare prefixes that aren't removed by light stemming like some prepositions that may be attached to the beginning of the word. This leads to think about using N-gram stemming after applying light stemmer. N-gram stemming has the ability to discover the similarity between words even if they are attached to any affixes. Table 5 shows an example for some groups of words clustered using the suggested hybrid approach of light and trigram stemming with similarity threshold (0.8).

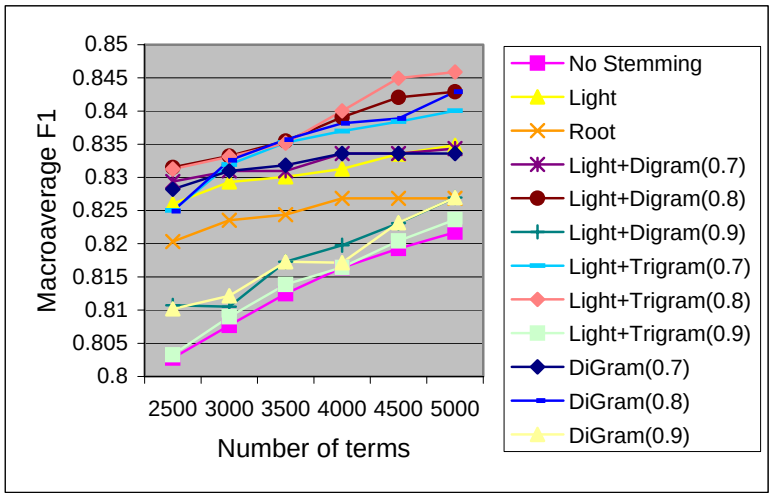

Figure 3: Effect of stemming in categorization accuracy.

Table 5: Some groups of words clustered using hybrid approach of light and tigram stemming with similarity threshold (0.8).

الاستثمار ، استثمار ، للاستثمار ، الاستثماريه ، الاستثماري ، والاستثمارات ، استثمارات ، بالاستثمار ، الاستثمارات ، والاستثماريه ، للاستثمارات ، باستثمارات ، والاستثمار ، والاستثماري ، استثماريه ، واستثمار ، استثماريا ، وباستثمارات
الاقتصاديه ، والاقتصادي ، الاقتصاد ، بالاقتصاد ، الاقتصادي ، والاقتصاد ، اقتصادي ، الاقتصاديين ، اقتصاديه ، والاقتصاديه ، واقتصادي ، الاقتصادات ، واقتصاديا ، اقتصاد ، اقتصاديا ، لاقتصاديات
الاجتماعي ، والاجتماعيه ، الاجتماع ، الاجتماعين ، بالاجتماع ، والاجتماعات ، الاجتماعيه ، للاجتماعات ، والاجتماعي ، الاجتماعات ، اجتماعي ، اجتماعيا ، واجتماعيه ، للاجتماع

5.3 Term Selection

After evaluating the stemming process, it is tried to find the best term selection criteria. After stemming, the dictionary of words is formed with thousands of terms, so that it is a must to find the most valuable set of terms by term selection. Section 4.2.1 discusses different term selection criteria. Document Frequency Thresholding, Information Gain, CHI Square, Odds Ratio, NGL Coefficient and GSS Score are evaluated by using the hybrid approach of light and trigram stemming with similarity threshold (0.8) in the stemming phase, boolean weighting for document representation and Rocchio classifier with β=1.6 and γ=0.4 for classification.

Experiments in figure 4 show that using any of those criteria separately gave a near results, also it is noticed that when using Information Gain and Odds Ratio, most of the documents don't contain any term in the list of the selected terms, in other words Information Gain and Odds Ratio select terms with rare appearance in the data set (i.e. terms with very low document frequency). This problem motivated the using of a hybrid approach between Document Frequency Thresholding other criteria. Document Frequency is used to remove rare terms and the other criteria to select terms from the remaining list.

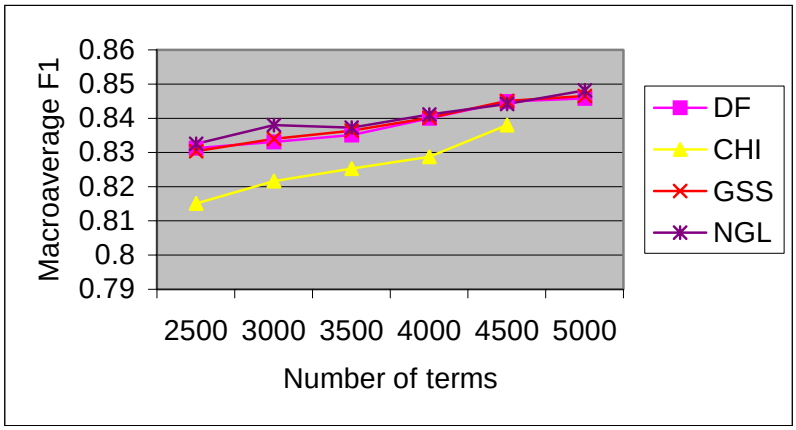

Figure 4: Effect of term selection criteria in categorization accuracy.

When using this hybrid approaches, results show that using Document Frequency Thresholding to remove terms with document frequency less than 2 then selecting terms that have high Information Gain score gave the highest results, when using Document Frequency Thresholding to remove terms with document frequency less than 3 the number of terms remains was about 4100 term so that this hybrid method is tested to select number of terms less than 4000 only, one draw back for this method is that some few documents when being represented as vectors, all their terms weight is zero (i.e. it contains no term from the selected list of terms). Figure 5 shows the accuracy of classification process for different hybrid term selection approaches.

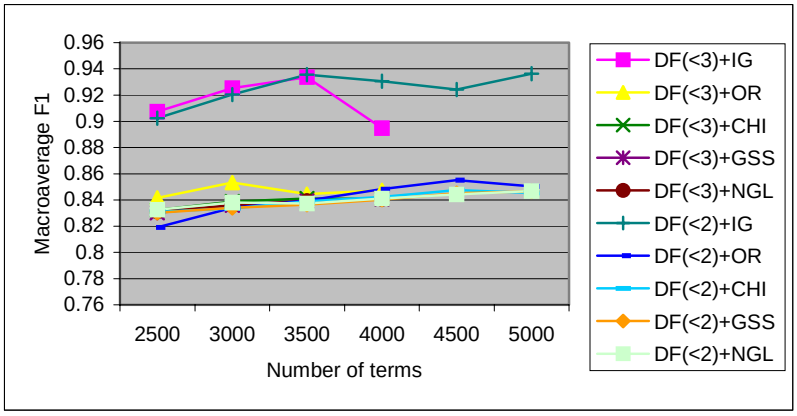

Figure 5**:** Effect of hybrid term selection criteria in categorization accuracy.

5.4 Term Weighting

After examining the best criteria for feature selection, the suitable term weighting method is examined. Traditional schemas like tf, tfidf, boolean, and normalized-tfidf methods described in section 4.2.2 are tried. In experiments, hybrid approach of light and trigram stemming with similarity threshold (0.8) is used for the stemming phase, hybrid feature selection criteria of Document Frequency Thresholding and Information Gain and Rocchio classifier with β=1.6 and γ=0.4 for classification. Results presented in figure 6 shows that normalized-tfidf is the preferable method for term weighting.

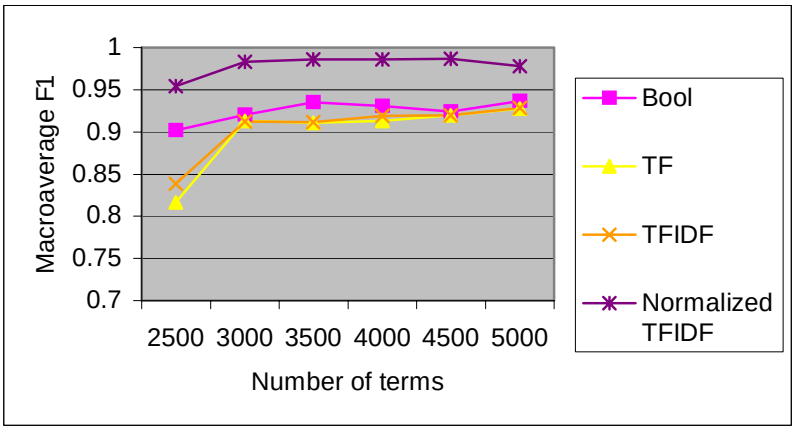

Figure 6: Effect of term weighting method in categorization accuracy.

5.5 Classifier

Finally, two non-parametric classification methods widely used with text categorization tasks were used, the k-NN classifier and the Rocchio classifier discussed in section 4.3.1 and 4.3.2 respectively. Hybrid approach of light and trigram stemming with similarity threshold (0.8) is used for the stemming phase, hybrid feature selection criteria of Document Frequency Thresholding and Information Gain and normalized-tfidf method for term weighting.

Table 6: Time in hours used for classifying the data corpus using K-NN and Rocchio classifiers.

No. of terms	K-NN	Rocchio
2500	0:50:04	0:00:16
3000	1:12:28	0:00:20
3500	1:36:43	0:00:22
4000	2:30:45	0:00:28
4500	3:57:27	0:00:30
5000	5:18:25	0:00:32

Results in figure 7 and table 6 show that Rocchio classifier is superior over k-NN classifier in both time and accuracy, different values for k (form k=1 to k=19), and for β and γ for Rocchio classifier are used. As discussed in section 5, k-NN has many disadvantages of selecting value for k, also k-NN is not efficient while Rocchio classifier is more efficient as it classifies documents using centroids of every class instead of using every training document in the data corpus. Best values for β and γ are 1.6 and 0.4 respectively.
As noticed, the error rate is very small, this results form using a small size data corpus. If the data corpus is huge enough, this may lead to increasing in the error rate.

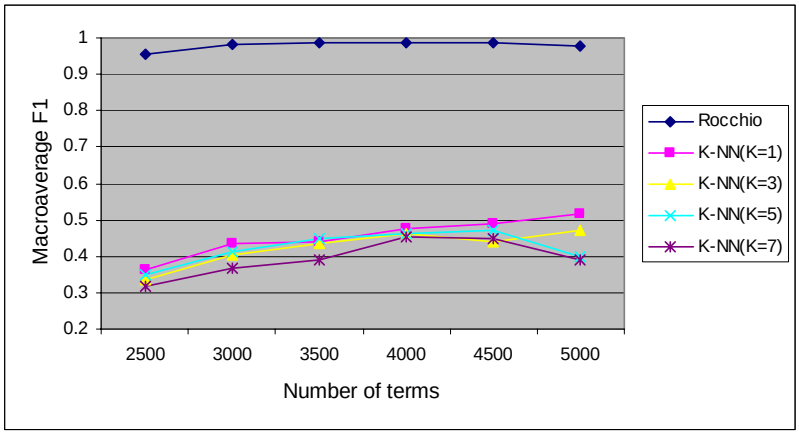

Figure 7: Effect of classifier in categorization accuracy

6. CONCLUSION AND FUTURE WORK

This paper presents an applying of machine learning strategies in the field of Arabic text categorization. Many literatures discussed text categorization systems for other languages but few researches presented for Arabic language according a performed survey. Text collection is collected from the local newspapers, then performing the preprocessing routines on the documents. This preprocessing includes removal of stop words and stemming the documents to cluster the terms according to their similarity. Three stemming approaches are being tested, results show that hybrid approach of light and statistical stemmer is the most suitable for text categorization task in Arabic language. After stemming, words' dictionary is constructed from terms that appear in all the documents at least once, due to the very high dimensionality of this dictionary, several methods for selecting highly informative terms are used. A hybrid method for term selection is proposed by combining Document Frequency Thresholding and Information Gain, this proposed method gives high results. After term selection, every document is represented as a vector of terms' weights. Four term weighing criteria are used, normalized-tfidf is the suggested weighting method. Finally, two non-parametric classifiers are used; the k-NN classifier and Rocchio classifier. Rocchio classifier shows superiority over k-NN in both efficiency and generalization. Thus this paper recommends the following structure for Arabic text categorization: using statistical n-gram stemmer for document preprocessing, hybrid approach of Document Frequency Thresholding and Information Gain for feature selection, normalized-tfidf for term weighting and Rocchio classifier for classification.

In the future we are looking to extend this work by doing some more preprocessing efforts needed specially for morphological language like Arabic like selecting terms locally from each class instead of selecting them globally from the whole corpus. Also the effect of feature extraction instead/beside term selection can be tested in the classification process, and finally, other classifiers like Naïve Bayes and Support Vector Machines (SVM) and neural networks can be used for classification process.

REFERENCES

[1] F. Sebastiani. “Machine learning in automated text categorization**”.** ACM Computing Surveys, volume 34 number 1. PP 1-47. 2002.

[2] M. Aljlayl and O. Frieder. “On Arabic search: improving the retrieval effectiveness via a light temming approach”. In ACM CIKM 2002 International Conference on Information and Knowledge Management, McLean, VA, USA. PP 340-347. 2002.

[3] S. Al-Fedaghi and F. Al-Anzi. “A new algorithm to generate Arabic root-pattern forms”. In proceedings of the 11th national Computer Conference and Exhibition. PP 391-400. March 1989.

[4] R. Al-Shalabi and M. Evens. “A computational morphology system for Arabic”. In Workshop on Computational Approaches to Semitic Languages, COLING-ACL98. August 1998.

[5] S. Khoja. “Stemming Arabic Text”. Lancaster, U.K., Computing Department, Lancaster University. 1999.

[6] L. Larkey, and M. E. Connell. “Arabic information retrieval at UMass in TREC-10”. Proceedings of TREC 2001, Gaithersburg: NIST. 2001.

[7] L. Larkey, L. Ballesteros, and M. E. Connell. “Improving Stemming for Arabic Information Retrieval: Light Stemming and Co-occurrence Analysis”. Proceedings of SIGIR’02. PP 275–282. 2002.

[8] A. Chen and F. Gey. “Building an Arabic Stemmer for Information Retrieval”. In Proceedings of the 11th Text Retrieval Conference (TREC 2002), National Institute of Standards and Technology. 2002.

[9] F. Sebastiani. “A Tutorial on Automated Text Categorisation”. Proceedings of ASAI-99, 1st Argentinian Symposium on Artificial Intelligence. PP 7-35. 1999.

[10] Y.Yang, and J. O. Pedersen. “A comparative study on feature selection in text categorization”. Proceedings of ICML-97. PP 412-420. 1997.

[11] M. Rogati and Y. Yang. “High-Performing Feature Selection for Text classification”. CIKM'02, ACM. 2002.

[12] T. Liu, S. Liu, Z. Chen and Wei-Ying Ma. “An Evaluation on Feature Selection for Text Clustering”. Proceedings of the 12th International Conference (ICML 2003), Washington, DC, USA. PP 488-495 . 2003.

[13] K. Aas and L. Eikvil. “Text categorisation: A survey”, Technical report, Norwegian Computing Center. 1999.

[14] S. H. Mustafa and Q. A. Al-Radaideh. “Using N-grams for Arabic text searching”. Journal of the American Society for Information Science and Technology Volume 55, Issue 11. PP 1002–1007 . 2004.

[15] T. Mitchell. “Machine Learning”. McGraw-Hill, New York. 1997.

[16] C. Liao, S. Alpha, and P. Dixon, "Feature preparation in Text Categorization", Australasian Data Mining Workshop in CEC 2003.

[17] J. Rocchio, “Relevance feedback in information retrieval”. In G.Salton (Ed.), The SMART Retrieval System: Experiments in Automatic Document Processing. PP 313-323. Prentice-Hall. 1971.

[18] W. W. Cohen and Y. Singer, “Context sensitive learning methods for text categorization”. ACM Transactions on Information Systems Volume 17, Issue 2. PP 141–173. 1999.

[19] R. Baeza-Yates and B. Ribiero-Neto, "Modern Information Retrieval", Addison Wesley, 1999.