CS 5604 : Information Storage and Retrieval

Instructor: Prof. Edward Fox

ProjArabic team

Classification of Arabic Documents

Project Final Report

By: Ahmed A. Elbery

12/11/2012

Table of Contents

Summary.....3
Background.....4
Arabic Text Classification: 6
Project Model.....8
Arabic documents collection.....8
Data pre-processing.....9
Classification 15
Results and evaluation..... 16
Conclusions &future work 19
References 20

Summary

Arabic language is a very rich language with complex morphology, so it has a very different and difficult structure than other languages. So it is important to build an Arabic Text Classifier (ATC) to deal with this complex language. The importance of text or document classification comes from its wide variety of application domains such as text indexing, document sorting, text filtering, and Web page categorization. Due to the immense amount of Arabic documents as well as the number of internet Arabic language users, this project aims to implement an Arabic Text-Documents Classifier (ATC).

Background

Text document classification (TC) is the process of assigning a given text to one or more categories. This process is considered as a supervised classification technique, since a set of labeled (pre-classified) documents is provided as a training set. The goal of TC is to assign a label to a new, unseen document [1].In order todetermine the appropriate category for an unlabeled text document,a classifier is used to perform the task of automatically classifyingthe text document. The rapid growth of the internet and computertechnologies has caused the existence of billions of electronic textdocuments which are created, edited, and stored in digital ways.This situation has brought great challenge to the public, specificallythe computer users in searching, organizing, and storing these documents.

TC is one of the most common themes in analyzing complex data. The study of automated text categorization dates back to the early 1960s. Then, its main projected use was for indexing scientific literature by means of controlled vocabulary. It was only in the 1990s that the field fully developed withthe availability of ever increasing numbers of text documents in digital form and thenecessity to organize them for easier use. Nowadays automatedTCis applied in a varietyof contexts - from the classical automatic or semiautomatic (interactive) indexingof texts to personalizedcommercials delivery, spam filtering, Web page categorizationunder hierarchical catalogues, automatic generation of metadata, detection oftext genre, and many others [2].

There are two mainapproaches to text categorization. The first is the knowledge engineering approach inwhich the expert's knowledge about the categories is directly encoded into the systemeither declaratively or in the form of procedural classification rules. The otheris the machine learning (ML) approach in which a general inductive process buildsa classifier by learning from a set of pre-classified examples.

TC may be formalized as the task of approximating the unknown target function $\Phi: D \times C \rightarrow \{T,F\}$ (that describes how documents ought to be classified,according to a supposedly authoritative expert) by means of a function$\Phi\hat{}: D\times C \rightarrow \{T,F\}$ called the classifier, where $C = \{c_1,...,c_n\}$ is a predefinedset of categories and D is a (possibly infinite) set of documents. If $\Phi(d_j,c_i) = T$,then d_j is called a positive example (or a member) of c_i, while if $\Phi(d_j,c_i) = F$ itis called a negative example of c_i [2].

Figure 1 is an example, where C = {Graphics,Arch,NLP,AI,Theory}, any document in the document space should be assigned to one of these categories by the function $\Phi\hat{}$

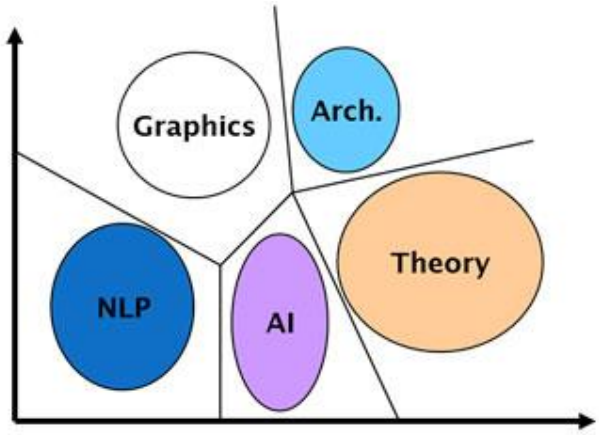


Figure1: TC example

Arabic Text Classification:

The importance of ATC comes from the following main reasons;

Due to Historical, Geographical, Religious reason; Arabic language is a very rich with documents.

A study of the world market, commissioned by the Miniwatts Marketing Group [3] shows that the number of Arab Internet users in the Middle East and Africa could jumped to 32 million in 2008 from 2.5 million in the year 2000, and in June 2012 this number jumped to more than 90 million users, the growth of Arab Internet users in the Middle East region (for the same period 2000-2012) is expected to reach about 2,640% compared to the growth of the world Internet users.

The conducted research pointed out that 65% of the Internet Arabic speaking users could not read English pages.

The big growth of the Arabic internet content in the last years has raised up the need for an Arabic language processing tools [4].

But on the other hand there are many challenges facing the development of Arabic language processing tools including ATC tools. The first is that Arabic is a very rich language with complex morphology. Arabic language belongs to the family of Semitic languages. It differs from Latin languages morphologically,syntactically and semantically. The writing system of Arabic has 25consonants and three long vowels that are written from right to left and change shapesaccording to their position in the word.

In addition, Arabic has short vowels (diacritics)written above and under a consonant to give it its desired sound and hence give a word adesired meaning. The common diacritics used in Arabic language are listed in Table 1 [5].

Table 1 Common short vowels (diacritics) used in Arabic text

Diacritic	Shape description	Example	Sound
fatha	A small diagonal line appears above a letter	دَ	da
kasra	A small diagonal line appears below a letter	دِ	di
damma	A small "comma-like" diacritic placed above a letter	دُ	du
tanwin	A double vowel diacritic appears only at the end	دً دٍ دٌ	dan, din, dun
sukun	A small circle shape above the letter indicating that the consonant is not followed by a vowel	دْ	d
shadda	A small diacritic (ّ) indicating a doubled consonant	دّ	dd
madda	A diacritic appears on top of alif indicating a long alif	آ	aa

The Arabic language consists of three types of words; nouns, verbs and particles. Nouns and verbs are derived from a limited set of about 10,000 roots [6].Templates are applied to the roots in order to derive nounsand verbs by removing letters, adding letters, or includinginfixes. Furthermore, a stem may accept prefixes and/orsuffixes in order to form the word [7]. So, Arabic language is highly derivative where tens or even hundreds of words could be formed using only one root. Furthermore, a single word may be derived from multiple roots [8].

In addition, it is very common to use diacritics (fully, partially even randomly) with classical poetry, children's literature and in ordinary text when it is ambiguous to read. For instance, a word in Arabic consisting of three consonants like (كتب ktb) "to write" canhave many interpretations with the presence of diacritics [9] such as shown in Table 2.

For Arabic language speakers, the only way to disambiguate the diacritic-less words isto locate them within the context. Analysis of 23,000 Arabic scripts showed that there is anaverage of 11.6 possible ways to assign diacritics for every diacritic-less word [10].

Table 2 Different interpretation of the Arabic word كتب (ktb) in the presence of diacritics

Arabic word	Transliteration	Part of speech	English meaning
كَتَبَ	*kataba*	3PSNG (verb)	Wrote
كُتُب	*kutub*	Noun	Books
كُتِبَ	*kutiba*	Passive (verb)	Written
كَتَّبَ	*kattaba*	Verb	Make someone to write

In addition to the above description of complex morphology, Arabic has very complex syntaxes, linguistic and grammatical rules.

It is clear that Arabic language has a very different and difficult structure than other languages. These differences makeit hard for language processing techniques made for other language to directly apply to Arabic.

So the objective of this project is to develop an Arabic Text-Document Classifier (ATC), and study the different techniques and parameters that may affect the performance of this classifier. In this project we will build ATC model and discus its implementation problems and decisions. Also we will use multiple classification techniques, namely support vector machine, Naive Bayes, k-Nearest Neighbors, and decision trees. We will compare between them from the accuracy and processing time perspectives. This project can be then used as a base for other students who want to continue in this field to make deeper studies.

Project Model

The project passes through three main phasesas shown in figure 2
Arabic documents collection
Data preprocessing
Classification

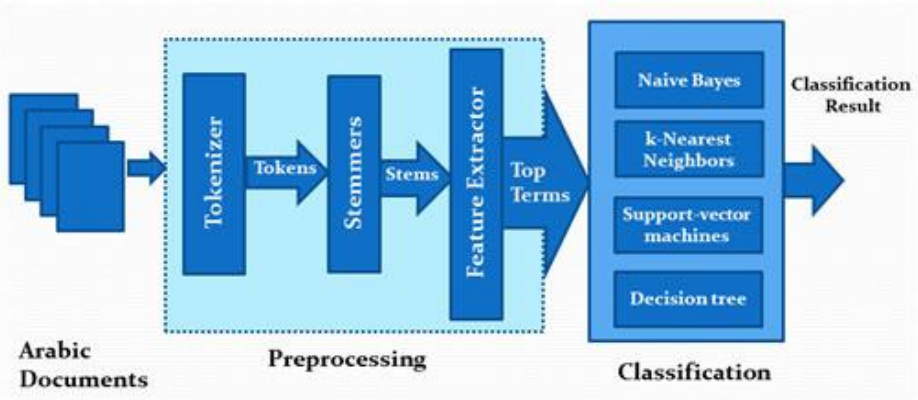


Figure 2: ATC model

Arabic documents collection.

In this phase, we collect the data set that will be used for building and testing the classifier module. At this point we had to make an important decision; whether we will deal with diacritics. As we mentioned earlier, diacritics is very important in Arabic documents, for instance to distinguish between the two words (ذَهَبَzhb) which means "to go" and the word (ذَهَبْ zhb) which means "gold" , the only way is to depend on diacritics. The two words are totally different; "Go" can appear in many contexts without identification for any topic an category, while the word "Gold", when it appears frequently in a document it means that this document could be categorized under financial or economical context.

But on the other hand working with diacritics in documents is very difficult, not only because it increases the character space from 28 letters to more 300 character.But also, diacritics are subjected to many complex Arabic grammars. So, if we want to consider the diacritics we have to consider the Arabic grammar and syntax which is a very complex problem.In the project,we used diacritic-lessdocuments or documents with very little diacritics.

Our document collection consists of one hundred documents all of them are about Arabic spring; these documents are categorized into two main topics; violence and politics in Arabic spring. Each category is 50 documents. We collected these documents from the Arabic news websites. For each of these categories there are many words that are expected to be

more frequent, for each category we expect to find a set of frequent words, as well, we expect to find a set of common words for both categories as shown infigure 3.

السياسية الحرية المنطقة القومي دولي الأنظمة جغرافي

Geographic Systems International National Area Liberty Politic

(a)

عنف قتل انفجار مواجهات ميليشيات سلاح حرق

Burn Weapon Militias Clashes Explosion Kill Violence

(b)

القوى قرار شارك حكومة الشعوب مسؤول

Responsible Peoples Government Share Decision Forces

(c)

Figure 3: (a) Some frequent words in politics documents, (b)Some frequent words in violence documents, (c)Some common in violence and politics documents

These documents are located into two subfolders called "Politics" and "Violence" in a "DATA" folder and the path of this folder is passed to the next phase to start working on these documents.

Data pre-processing

In this phase, documents are processed and prepared to be used by the classification phase. This phase has three main sub-phases; Tokenizer, Stemmer, and Feature extractor

Tokenization

Tokenizer is responsible for scanning each document word by word, and extracting words in this document. It has two main steps; tokenization and text cleaning.

In the tokenization, the Arabic Tokenizer uses White Space Tokenization because the space is the only way to separate words in Arabic language, i.e. dash and hyphen are not used to separate words in Arabic. Then in the text cleaning step, it removes the non-Arabic letters, numbers and punctuations as shown in Figure 4.

Figure 4: Tokenization example

It also removes the stop works such as stop words pronouns, conjunctions,and prepositions. As well, it removes numbers, and names.

In Arabic, identifying and removing names is not an easy task like that in other languages.In English for example, the capital letter are used for identifying names and abbreviations in the middle of sentences, while in Arabic, the concept of capital letters does not exist at all. Another problem is that most Arabic names actually come from verbs and could be stemmed to regular stems, also Arabic names and could be used as adjectives or other parts of the sentences. The most suitable technique for identifying names may be based on sentence analysis. But the problem facing these techniques is the complexity of Arabic. i.e. the basic sentence structure for Arabic is very flexible and may take the form of "Subject Verb Object" , "Verb Subject Object" or "Verb Object Subject", the basic Arabic sentence can take any of these three forms. This is a simple example of the complexity of Arabic sentence structure.

The simplest way to detect names in a document is to use a name list, and when the Tokenizer extracts a word it compares it against this list. But this solution is not effective since we need to add all names in this list, as well; it cannot deal with the compound names.

Stemming

The main goal of the stemming is toimprove the efficiency of the classification by reducing the number of terms being input to the

classification. As we mentioned earlier, Arabic language is highly derivative where tens or even hundreds of words could be formed using only one stem, furthermore, a single word may be derived from multiple stem. So, working with Arabic document words without stemming results in an enormous number of words are being input to the classification phase. This will definitely increase the classifier complexity and reduce its scalability.

Many stemming methods havebeen developed for Arabic language. These stemmers are classified intotwo categories. The first one is root extraction stemmerlike the stemmer introduced by [11]. The secondis light stemmers like the stemmer introduced by in [12].

In this project we used a Rule-Based Light Stemmer introduced in [13]. In this stemmer, to solves the problem of prefix/suffix sequenceambiguity, words are firstly matched against a set of all possible word patterns in Arabic before prefix/suffix truncation, so if a word starts with apossible prefix but itmatched one of the possible patterns, then it's a validword and this prefix is part of the originaland should not be truncated.

Then, if the word didn't match any of thepatterns, then the compatibilitybetween the prefix and suffix should be found, where some suffixescould not be combined with certain suffixes in thesame word. If the prefix and suffix are combatable then they could be removed from this word. For example the prefix "ال"may not be combined with the suffix "ك" so we cannot say "الكتابك" and thus if we have a word like"الكرنك" the stemmer will not remove the prefix and suffixwhich lead to the wrong word "كرن" but it willdetect that the last character "ك" is part of theoriginal word and not a suffix, and soit will onlyremove the prefix "ال" which will lead to the correctstem "كرنك"

If the combination of the prefix and suffix is validthen the stemmer counts the letters of the word after removingthe prefix and suffix since Arabic words other thanconjunctions like "من" ،"في" consists of at least 3characters. Based on this count it will take the prober decision.

Finally, the stemmer tries to solve the problem of so called broken plural in Arabic, in which, a noun in pluraltakes another morphological form different from its initial form in singular. To do that, the stemmer keeps a table of patterns for all broken plural and their singular form, this table is shown in Table 3.

Table 3 Singular and plural patterns.

Plural Pattern	Singular Pattern
مفاعل	مفعل
مفاعيل	مفعول
أفعال	فعل
فعلاء	فعل، فعال، فاعل، فعيل
فعال	فاعل
أفعل	فعل
أفعلة	فعيل، فعال
فواعل	فوعل، فاعل

The following (figure 5) is the result of the stemming phase in our project for the terms shown in figure 4.

غور - جغرافيا - سوس - فعل - ربع - عرب - حدث - دخل - كبر - قود - غور - جذر - جراً - بلد - كتل - شرق - الآن - دنج - سمي - ربع - عرب - غور - دخل - بلد - عرب - الت - غور - لوح - دول - قلم - خوص - نطق - شرق - وسط

Figure 5: The stemmer output

Feature extraction

This phase starts by splitting the data set into training set and test set. The size of the test set is determined by a parameter "test_set_ratio".

In this sub-phase, the most informative terms are extracted from documents. There are two main benefits from the feature extraction. The first is that it reduces the number of dimensions(terms) and thus reduces the classifier complexity and processing requirements (Time, Memory, & Desk Space). The second is increasing the classification efficiency by removing noise features and avoid over fitting caused by terms that are frequent in all categories. The extraction is supported by the experiment conducted in [14], which illustrates that selecting 10% of features exhibits the same classification performance as when using all the features when using SVM in classification. Based on this argument we use a parameter in the feature selection phase called "feature_ratio" and set it to 10%.

The feature selection can be based on many parameters such as tf.idf, Correlation Coefficient Chi2, and information Gain (IG). In this project we used the tf.idf as selection criteria since it also removes terms which are frequent in all classes thus reducing the over fitting.

Figure 6-a shows an example of 4 documents (2 politics + 2 violence) and the calculated values for the feature extraction phase, assuming these documents are input to the feature extraction phase.The classifier selects the second document as a test set and the other three as training set. First the term count for each term is calculated, then the term frequency. Then the document frequency and the inverse document frequency (log3/df) are

calculated. Finally the tf.idf is calculated as shown in figure 6-b. then to find the highest weighted terms, we sum the tf.idf of each term and the terms with the highest values are selected. In this example we configured the feature_ratio to 50%. The selected features shown in figure 6-c are then passed to the next phase with their classes.

The same calculations are made for the test set except that for the test set we donot need to find the tf.idf for all terms but only for the selected terms

(a)

		'Burn'	'Government'	'International'	'Kill'	'Liberty'	'Militias'	'Politics'	'Systems'	'Violence'	'Weapon'	'area'	'nation'
Term count	Doc P-1	0	1	1	0	1	0	2	1	0	0	1	1
	Doc V-1	1	1	0	1	0	1	0	1	2	2	0	0
	Doc V-2	1	1	0	2	0	1	0	1	1	1	0	0
tf	Doc P-1	0	0.125	0.125	0	0.125	0	0.25	0.125	0	0	0.125	0.125
	Doc V-1	0.1111	0.1111	0	0.1111	0	0.1111	0	0.1111	0.2222	0.2222	0	0
	Doc V-2	0.125	0.125	0	0.25	0	0.125	0	0.125	0.125	0.125	0	0
df		2	3	1	2	1	2	1	3	2	2	1	1
idf		0.4055	0	1.0986	0.4055	1.0986	0.4055	1.0986	0	0.4055	0.4055	1.0986	1.0986
tf.idf	Doc P-1	0	0	0.1373	0	0.1373	0	0.2747	0	0	0	0.1373	0.1373
	Doc V-1	0.0451	0	0	0.0451	0	0.0451	0	0	0.0901	0.0901	0	0
	Doc V-2	0.0507	0	0	0.1014	0	0.0507	0	0	0.0507	0.0507	0	0
sum (tf.idf)		0.0957	0	0.1373	0.1464	0.1373	0.0957	0.2747	0	0.1408	0.1408	0.1373	0.1373

(b)

	'International'	'Kill'	'Liberty'	'Politics'	'Violence'	'Weapon'
Doc P-1	0.137327	0	0.137327	0.274653	0	0
Doc V-1	0	0.045052	0	0	0.090103	0.090103
Doc V-2	0	0.101366	0	0	0.050683	0.050683

(c)

Figure 6: Feature extractionexample

Classification

In this phase we use 4 classification algorithms, for each of them, the features extracted from the training set in the preprocessing phase are used to the classification algorithm to build the classification model, and then the weights of the test set are used by this model to find the classes for this set as shown in figure 7.

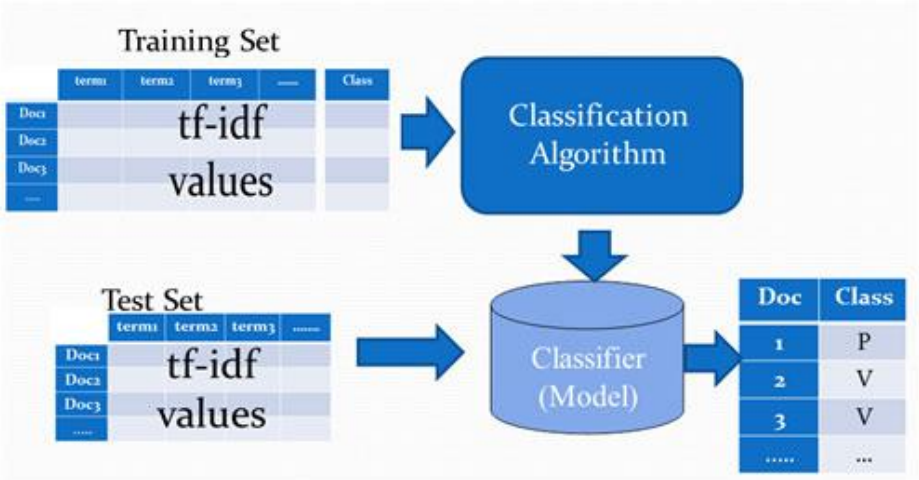


Figure 7:the classification phase

In the project we tested four classification algorithms, support vector machine, Naive Bayes, k-Nearest Neighbors, and decision trees, and we then calculated the accuracy of each of them, also we calculated the processing time required for the model building and the classification steps for each algorithm.

Results and evaluation

To compare the 4 algorithms, we run the algorithm 10 times on the data set, with test set ratio 20% and feature ratio 10% and calculate the accuracy and the processing time for each algorithm.

Figure 8 shows the average accuracy of the 4 algorithms for the ten runs. It is clear that the SVM has the best accuracy among the four algorithms.

Figure 9-a shows the accuracy for each algorithm in each run, and figure 9-b shows the correlation coefficient for them. From this figure 9-b we conclude that the SVM and KNN positively correlated, this means that they behave similarly when the data set changes.

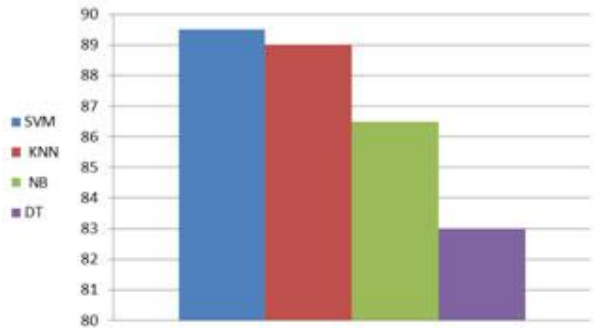
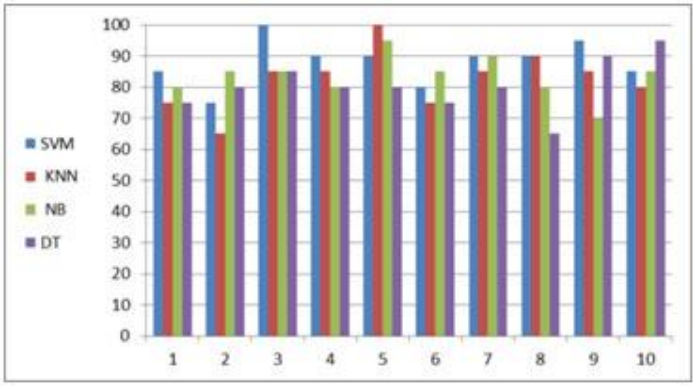


Figure 8: the average accuracy



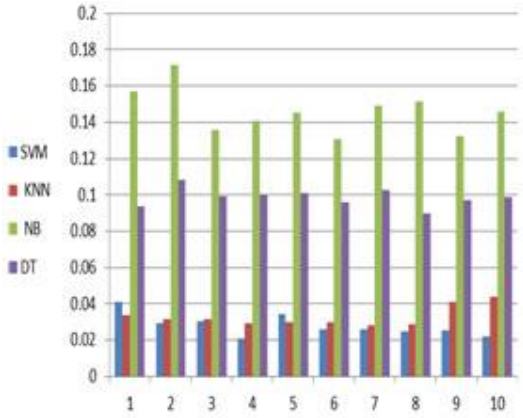
(a)

	SVM	KNN	NB	DT
SVM	1			
KNN	0.695182	1		
NB	-0.18592	0.240441	1	
DT	0.205563	-0.05273	-0.08491	1

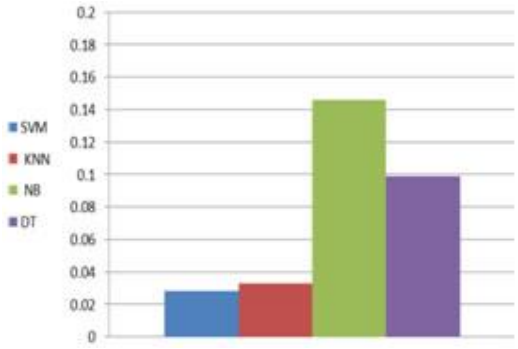
(b)

Figure 9: (a) the accuracy, (b) The correlation coefficient between algorithms

Regarding the processing time, Figure 10 shows the average processing time for each of the four algorithms, and it is clear that SVM is has the smallest processing time.



(a)



(b)

Figure 10: (a) the processing time, (b) the average processing time

Conclusions &future work

From the results we can conclude that the SVM has better performance in both time and accuracy. This project could be the base for many other future work, Other students can use this model to check the effect of the feature ratio on the performance by studying its effect on both time and accuracy. It is also beneficial to compare different selection parameters such as chi2 or information gain to find whether changing the selection parameter affects and how this effect can be used to enhance the classifier performance.In addition, deep study of each algorithm can be conducted by changing its parameters. i.e. in this project we use K=1 for the KNN, so other students can study the effect of increasing this parameter on the classifier performance.

References

[1] Sebastiani,. Machine learning in automated text categorization. ACM Computing Surveys (CSUR), 2002, 34(1), 1-47.

[2] Ronen Feldman, James Sanger . The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data, 2007

[3] http://www.internetworldstats.com

[4] Aitao C., "Building an Arabic Stemmer for Information Retrieval," in Proceedings of the Eleventh Text Retrieval Conference, Berkeley, pp. 631-639, 2003.

[5]Hammo, B. H. (2009, this issue). Towards enhancing retrieval effectiveness of search engines for diacritisized Arabic documents. *Information Retrieval*.

[6] Building a shallow Arabic morphological analyzer in one day. In Proc. Of the 40th Annual Meeting of the Association for Computational Linguistics (ACL'02) (pp. 1-8).

[7] Darwish, K. (2003). *Probabilistic methods for searching OCR-degraded Arabic text.* Ph.D. Thesis, Electrical and Computer Engineering Department, University of Maryland, College Park.

[8] Ahmed, Mohamed Attia, "A Large-Scale Computational Processor of the Arabic Morphology, and Applications." A Master'sThesis, Faculty of Engineering, CairoUniversity, Cairo, Egypt, 2000.

[9] Kirchhoff, K., & Vergyri, D. Cross-dialectal data sharing for acoustic modeling in Arabic speech recognition. Speech Communication, 46(1), 37-51, 2005.

[10] Debili, F., Achour, H., & Souissi, E. Del'etiquetage grammatical a' la voyellation automatique de l'arabe. Correspondances (Vol. 71, pp. 10-28). Tunis: Institut de Recherche sur le Maghreb Contemporain.

[11] Khoja S. and Garside R., Stemming Arabic Text, available at: http://www.comp.lancs.ac.uk/computing/users/khoja/stemmer.ps, last visited 1999.

[12] Leah L., and Lisa B., and Margaret C., Light Stemming for Arabic Information Retrieval, University of Massachusetts, Springer, 2007.

[13] G. Kanan and R. Al-Shalabi, "Building an Effective Rule-Based Light Stemmer for Arabic Language to Improve Search Effectiveness", IEEE, pp. 312-316, 2008.

[14] Debole, F. & Sebastiani, F. (2005). An analysis of the relative hardness of reuters-21578 subsets. Journal of the American Society for Information Science and Technology (JASIST), 56(6), 584- 596.