Arabic supervised learning method using N-gram

Majed Sanan
Paris 8 University, Paris, France
Mahmoud Rammal
Lebanese University, Beirut, Lebanon, and
Khaldoun Zreik
Paris 8 University, Paris, France

Abstract

Purpose – Recently, classification of Arabic documents is a real problem for juridical centers. In this case, some of the Lebanese official journal documents are classified, and the center has to classify new documents based on these documents. This paper aims to study and explain the useful application of supervised learning method on Arabic texts using N-gram as an indexing method ($n = 3$).
Design/methodology/approach – The Lebanese official journal documents are categorized into several classes. Supposing that we know the class(es) of some documents (called learning texts), this can help to determine the candidate words of each class by segmenting the documents.
Findings – Results showed that N-gram text classification using the cosine coefficient measure outperforms classification using Dice's measure and TF*ICF weight. Then it is the best between the three measures but it still insufficient. N-gram method is good, but still insufficient for the classification of Arabic documents, and then it is necessary to look at the future of a new approach like distributional or symbolic approach in order to increase the effectiveness.
Originality/value – The results could be used to improve Arabic document classification (using software also). This work has evaluated a number of similarity measures for the classification of Arabic documents, using the Lebanese parliament documents and especially the Lebanese official journal documents Arabic corpus as the test bed.

Keywords Classification, Learning methods, Languages, Text retrieval, Lebanon

Paper type Research paper

1. Introduction

The rapid growth of the internet has increased the number of online documents available. This has led to the development of automated text and document classification systems that are capable of automatically organizing and classifying documents. Text classification (or categorization) is the process of structuring a set of documents according to a group structure that is known in advance. There are several different methods for text classification, including statistical-based algorithms, Bayesian classification, distance-based algorithms, *k*-nearest neighbors, decision tree-based methods, etc.

Laila KHREISAT, 2006

Text classification techniques are used in many applications, including e-mail filtering, mail routing, spam filtering, news monitoring, sorting through digitized paper archives, automated indexing of scientific articles, classification of news stories, and searching for interesting information on the web.

The majority of these systems is designed to handle documents written in non-Arabic language, developing text classification systems for Arabic documents is a challenging task due to the complex and rich nature of the Arabic language. The Arabic language consists of 28 letters. The language is written from right to left. It has very complex morphology, and the majority of words have a tri-letter root. The rest have either a quad-letter root, penta-letter root, or hexa-letter root.

Interactive Technology and Smart Education
Vol. 5 No. 3, 2008
pp. 157-169

1741-5659
DOI 10.1108/17415650810908249

In our approach, we will use only the similarity measures and compare the results in order to know the convenient measure in classification using N-grams. And because that classification is one method of text mining we will explain in the following paragraph the steps of text mining, then we will see the preprocessing and indexing of texts before to be classified. At the next paragraph, we will explain the different similarity measures that we will use in our approach, and then the effectiveness measure used to calculate the precision and recall of each class. At paragraph 6 we will explain our approach and experiments, and finally we will see the conclusion and future approaches.

2. Text mining

2.1 Definition

Text mining is defined[1] as the non-trivial extraction of implicit, previously unknown, and potentially useful information from (large amount of) textual data.

Text mining is the process of applying automatic methods to analyse and structure textual data in order to create useable knowledge from previously unstructured information.

WIKIPEDIA (modifié depuis)

2.2 Text mining methods

There are many text mining applications or methods. Four of these methods are the following:

- information retrieval (IR);

This method consists of indexing and retrieval of textual documents.

- information extraction;

It means extraction of partial knowledge in the text.

- web mining;

It consists on indexing and retrieval of textual documents and extraction of partial knowledge using the web.

- classification;

Given: a collection of labelled documents (training set), the goal is to find a model for the class as a function of the values of the features.

2.3 Text mining steps

These steps concern principally the manner in which a text is represented (or structured), the choice of predicted algorithm to use, and then how to evaluate the obtained results to guarantee a good generalization of the model applied.

2.4 Representation of the information

In this step, we have to segment the unstructured information and put the units segmented into a table. But we have to choose the descriptors (important terms in documents) which can be chosen as words, lemmas, stemmas, or *N*-grams (characters or words or phrases).

And finally in some cases we have to think how to reduce the dimension of this textual space.

2.5 Automatic categorization of documents

This is the second step, the text categorization can be defined as the process that permit to associate a category(ies) or class(es) to a text (or document), in function of information contained in this text .

This association is very long and expensive then we think about the automation of this process.

The functional link between a class and a document, that is called "a prediction model", is estimated by a machine learning method.

The categorization of documents comports a choice of a learning technique (or classifier). The main classifiers used are the following:

- discriminated factorial analysis (Lebart and Salem, 1994);
- neuronal network (Wiener *et al.*, 1995; Schütze *et al.*,1995; Stricker, 2000);
- *K*-neighbors (Yang and Chute, 1994; Yang and Liu,1999);
- decision tree (Lewis and Ringuette, 1994; Apté *et al.* 1994); and
- Bayesian network (Borko and Bernick, 1964; Lewis, 1998; Androutsopoulos *et al.*, 2000; Chai *et al.*, 2002; Adam *et al.*, 2002).

Thèse Radwan JALAM, 2006

2.6 Validation method

In this final step, we have to evaluate the obtained results to guarantee a good generalization of the model applied.

3. Indexing

All text documents went through a preprocessing stage. This was necessary due to the variations in the way text can be represented in Arabic. The preprocessing was performed for the documents to be classified and the training classes themselves. Preprocessing consisted of the following steps:

Laila KHREISAT, 2006

- convert text files to UTF-8 encoding;
- remove punctuation marks, diacritics, non-letters, stop words. The definitions of these were obtained from the Khoja stemmer;
- replace initial آ إ, with أ with ا., and
- replace final ى followed by ء with.

XU, FRASER, WEISCHEDEL, 2002

3.1 Spelling normalization and mapping

Arabic orthography is highly variable. For instance, changing the letter YEH (ي) to ALEF MAKSURA (ى) at the end of a word is very common (Not surprisingly, the shapes of the two letters are very similar.). Since variations of this kind usually result in an "invalid" word, in our experiments we detected such "errors" using a stemmer (the Buckwalter Stemmer) and restored the correct word ending.

A more problematic type of spelling variation is that certain glyphs combining HAMZA or MADDA with ALEF (e.g. إ, أ, and آ) are sometimes written as a plain ALEF (ا), possibly because of their similarity in appearance. Often, both the intended word and what is actually written are valid words.

This is much like confusing "résumé" with "resume" in English. Since both the intended word and the written form are correct words, it is impossible to correct the spellings without the use of context.

We explored two techniques to address the problem.

(1) With normalization technique, we replace all occurrences of the diacritical ALEFs by the plain ALEF.

(2) With the mapping technique, we map a word with the plain ALEF to a set of words that can potentially be written as that word by changing diacritical ALEFs to the plain ALEF. In this absence of training data, we will assume that all the words in the set are equally probable.

Both techniques have pros and cons. The normalization technique is simple, but it increases ambiguity. The mapping technique, on the other hand, does not introduce additional ambiguity, but it is more complex.

3.2 Arabic stemming

Arabic has a complex morphology. Most Arabic words (except some proper nouns and words borrowed from other languages) are derived from a root. A root usually consists of three letters. We can view a word as derived by first applying a pattern to a root to generate a stem and then attaching prefixes and suffixes to the stem to generate the word (Khoja and Garside, 1999). For this reason, an Arabic stemmer can be either root-based or stem-based.

XU, FRASER, , WEISCHEDEL, 2002

3.3 Character N-grams

Broken plurals are very common in Arabic. There is no existing rule-based algorithm to reduce them to their singular forms, and it seems that it would be not be straight-forward to create such an algorithm. As such, broken plurals are not handled by current Arabic stemmers.

One technique to address this problem is to use character N-grams. Although broken plurals are not derived by attaching word affixes, many of the letters in broken plurals are the same as in the singular forms (though sometimes in a different order). If words are divided into character N-grams, some of the N-grams from the singular and plural forms will probably match.

This technique can also handle words that have a stem but cannot be stemmed by a stemmer for various reasons. For example, the Buckwalter stemmer uses a list of valid stems to ensure the validity of the resulting stems. Although the list is quite large, it is still not complete. N-grams in this case provide a fallback where exact word match fails.

In previous work (Mayfield *et al.*, 2004), experiments have been made with N-grams created from stems as well as N-grams from words. N-grams were created by applying a shifting window of n characters over a word or stem. If the word or stem has fewer than n characters, the whole word or stem was returned. The following table shows some results of these experiments. Two methods of creating N-grams were tried: from words and from stems. Retrieval scores in Table I show that stem-based N-grams are better than word-based N-grams for retrieval. The probable reason is that some of the

Table I.
Retrieval results using N-grams

	Bigrams	Trigrams	Tetragrams
Words	0.1461	0.2990	0.2900
Stems	0.1655	0.3365	0.3165

Fuchun PENG, 2003

word-based N-grams are prefixes or suffixes, which can cause false matches between documents and queries

However, character level N-gram models offer the following benefits and have been successfully used in many IR problems:

(1) Language independence and simplicity: character level N-gram models are applicable to any language, and even non-language sequences such as music or gene sequences.

(2) Robustness: character level N-gram models are relatively insensitive to spelling variations and errors, particularly in comparison to word features.

(3) Completeness: the vocabulary of character tokens is much smaller than any word vocabulary and normally is known in advance. Therefore, the sparse data problem is much less serious in character N-gram models of the same order.

4. Vector space model similarity and matching measures

In this paragraph, we will explain some similarity and matching measures in vector space model often used in IR, but we will use them in similarity between a document and a class.

Now that we have the document in a form that minimizes the information we need to consider when matching documents to classes we have to do some matching.

Under the vector and probabilistic models, the document is initially indexed in the same way as the classes.

*4.1 TF*ICF weight*

In fact I have used the TF*ICF and apply it to the class, then the query will be replaced by the document to be classified and the document will be replaced by the class, then I have defined the new weight TF*ICF.

In TF*ICF, ICF stands for inverse class frequency and TF stands for term frequency ("*" indicates multiplication).

The term frequency in the given class is simply the number of times a given term appears in that class. This count is usually normalized to prevent a bias towards longer classes (which may have a higher term frequency regardless of the actual importance of that term in the class) to give a measure of the importance of the term t_i within the particular class.

WIKIPEDIA (modifié depuis)

$$\mathrm{tf}_i = \frac{n_i}{\sum_k n_k}$$

where n_i is the number of occurrences of the considered term, and the denominator is the number of occurrences of all terms.

The inverse class frequency is a measure of the general importance of the term (obtained by dividing the number of all classes by the number of classes containing the term, and then taking the logarithm of that quotient).

$$\mathrm{ICF}_i = \log \frac{|C|}{|\{c : ti \in c\}|}$$

with $|C|$, total number of classes in the corpus and $|\{c : ti \in c\}|$, number of classes where the term t_i appears (that is $n_i \neq 0$).

Then

$$\text{TF ICF} = \text{TF}^{*}\text{ICF}$$

A high weight in TF–IDF is reached by a high-term frequency (in the given document) and a low document frequency of the term in the whole collection of documents; the weights hence tend to filter out common terms.

ICF is defined as log (no of classes in the collection/no of classes containing this document in the collection). This reflects the fact that uncommon documents are more likely to useful in narrowing down the selection of class than very common documents. TF is defined as log (frequency of term in this class) this reflects the fact that if a keyword occurs multiple times in a class, that class is more likely to be relevant than a class where the keyword occurs just once.

Prakash NADKARNI (support de cours, Yale Univ.

4.2 Coefficient of Dice

Now that the documents are both represented as vectors, the vector space model considers the similarity of them to be based on the angle between the two vectors in space. Up until this point (and with the probabilistic model), the vector has simply been a convenient mathematical model for storing a list of terms and their weights. The vector space model then makes the jump to processing them as if they were real geometrical vectors in a space with thousands of dimensions. Although this seems rather strange initially, it is based on an extension of a simple matching routine for non-weighted indexes. Consider non-weighted indexes for the above document and the sample class, these are basically a list of four words for the document and 14 for the class. A rough measure of matching strength is the number of terms they have in common, in this case two. This does not take into account how large each class is and would have a tendency to match larger documents, so we could divide by the number of terms in total between the document and the class. This leads to Dice's coefficient:

Mark D. DUNLOP, 1994

$$m = 2 * \frac{|D \cap C|}{|D| + |C|}$$

where $|D \cap C|$ is the number of terms common to the document and class, $|D|$ is the number of terms in the document, $|C|$ the number in the class, and m the matching value – the fraction is doubled to give a maximum value, for matching a class with itself, of 1 instead of 0.5.

4.3 Cosine coefficient

When considering weighted terms, like those we indexed, it is not possible to simply count the number of terms in common. Instead the vector space model multiplies the term weights together. For the vast majority of terms either the document or the class will have a zero weight, hence the resulting weight is zero. These individual new weights are then summed to give the top line of the matching algorithm. For a document D of N terms and for a class vector C, this leads to:

$$m = \frac{\sum_{i=1}^{N} D_i C_i}{\text{PDP.PCP}}$$

where $\|D\|$ is the length of the document (= total number of terms in document D), D_i is the weight of term i in vector D, and N is the total number of individual terms (the dimensionality of C). In geometry, this equation is used to calculate the cosine of the angle between the two vectors, hence this matching routine is known as the cosine coefficient.

Although quite simple to understand this approach has no sound bases in information theory there is no theoretical reason for this to be a good matching algorithm. The cosine coefficient does, however, perform well in practice, is reasonably easy to code and is used in many retrieval systems.

5. Precision and recall measures

Precision and recall are defined in Abdelali (2004) as follows:

Laila KHREISAT, 2006

$$\text{Precision} = \frac{\text{CC}}{\text{TCF}}$$

$$\text{Recall} = \frac{\text{CC}}{\text{TC}}$$

where CC, number of correct categories (classes) found; TCF, total number of categories found; TC, total number of correct categories.

Ever since the 1960s IR effectiveness is evaluated using the twin measures of recall and precision (Lavrenko, 2005).

To combine the two measures (precision and recall) in a single value, the F-measure is often used. The F-measure reflects the relative importance of recall versus precision. When as much importance is granted to precision as it is to recall we have the F_1-measure which is an estimation of the breakeven point where precision and recall meets if classifier parameters are tuned to balance precision and recall.

Then we can also use a single-number measure for the effectiveness as follows:

SOUCY et MINEAU, 2005

$$F_1 = \frac{2PR}{(P + R)}$$

where F_1 as a harmonic mean of precision and recall (Lavrenko, 2005).

For this study, relevance has been defined conceptually as:

6. Implementation

6.1 Corpus

We work on the meeting minutes of Lebanon parliament. Our database content all minutes from 1922 until 2005. The meetings are riche in different kind of information (economical, political, juridical, social, etc.). In other ways, we have a database for the official journal that content all laws and decrees. The texts of official journal are classified manually. Our work is to apply classification of official journal to parliament minutes in which Lebanese official journal documents form the main part. Then our corpus is the Lebanese official journal documents for year 2002 that form about 2,667 classified (labelled) documents.

6.2 Methodology

In our approach, we have chosen the N-gram method to represent information. And then to categorize the texts (Figure 1) we will use the learning method in which we

Radwan JALAM, thèse 2003

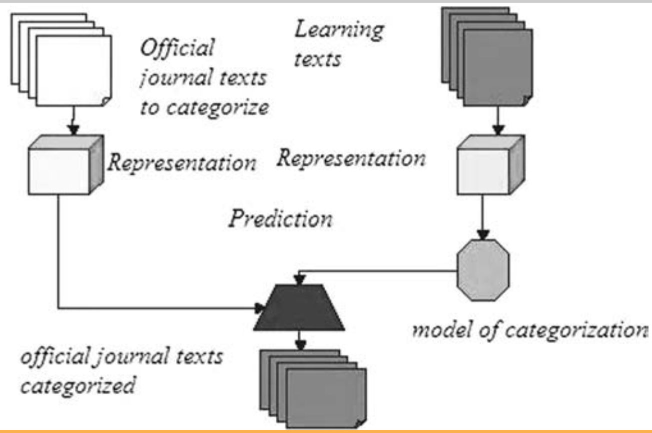

Figure 1.
Text categorization process

supposed that we have some categorized texts (learning texts) that we used to find the prediction method using the *N*-gram technique.

In our experiment, the learning texts will be the 2,667 Lebanese official journals (for the year 2002).

These documents are classified and each document belongs to one or multiple predefined classes.

We have two levels of classification, then each document belongs to a class of Level 1 and then to a subclass of a Level 2.

In general, we have three main classifications (Level 1):

- administrative classification;
- juridical classification; and
- thematic classification.

And in each classification we have different classes, for example in administrative classification we have 137 classes. Table II shows some of these classes.

Using these classified documents we will segment them by using N-gram method (three characters) (Figure 2) and then segment the two level classes.

Then we will try to find the candidate words for each class (Levels 1 and 2) and for each document using the vector space model.

After that we will apply the similarity and matching measures on documents and classes to classify automatically the pre classified documents.

Then we will conclude which is the convenient measure using the precision and recall parameters. Knowing the convenient measure we can then in the future use it to classify new documents.

6.3 Experimental software

To segment the 2,667 documents that form the corpus, using the N-gram method, I have made a program (using VB.net) that use the N-gram with 3,4, and 5 characters and give the result in a table (top 50).

6.4 Experience

Generating the N-gram profile consisted of the following steps:

Table II.
Administrative classes

ID	Description
1	مجلس الخدمة المدنية
3	المجلس العدلي
4	رئاسة مجلس الوزراء
5	المجلس الإسلامي العلوي
7	الهيئة العليا للإغاثة
9	مجلس الجنوب
10	المصرف الوطني للإنماء الصناعي و السياحي
11	وزارة العدل
13	المديرية العامة للتعليم العالي
14	المديرية العامة لرئاسة مجلس الوزراء
16	المديرية العامة للطرق و المباني
18	المديرية العامة لوزارة العمل
20	وزارة الخارجية و المغتربين
22	المديرية العامة للأحوال الشخصية
24	وزارة الزراعة
27	المديرية العامة للبريد

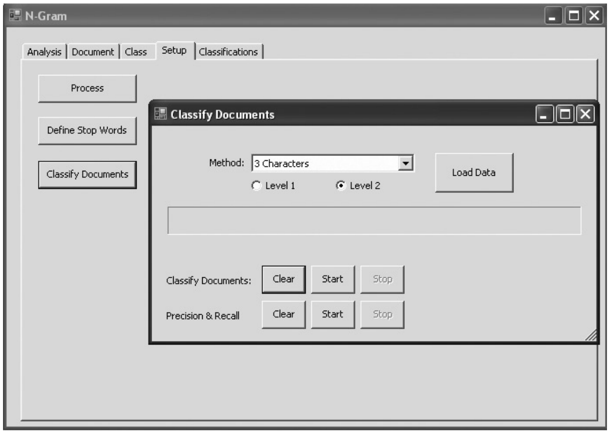

Figure 2.
Classify documents using trigrams

(1) Split the text into tokens consisting only of letters. All digits are removed.

(2) Compute all possible N-grams, for $n = 3$ (Trigrams).

Laila KHREISAT,2006

(3) Compute the frequency of occurrence of each *N*-gram.

(4) Sort the N-grams according to their frequencies from most frequent to least frequent. Discard the frequencies

(5) This gives us the N-gram profile for a document. For training class documents, the N-gram profiles were saved in text files. Each document to be classified went through the text preprocessing phase, and then the N-gram profile was generated as described above. The N-gram profile of each text document (document profile) was compared against the profiles of all documents in the training classes (class profile) in terms of similarity. Specifically, two measures were used.

6.5 Results

Calculate the precision and recall basing on similarity methods and then choose the convenient method (Figure 3).

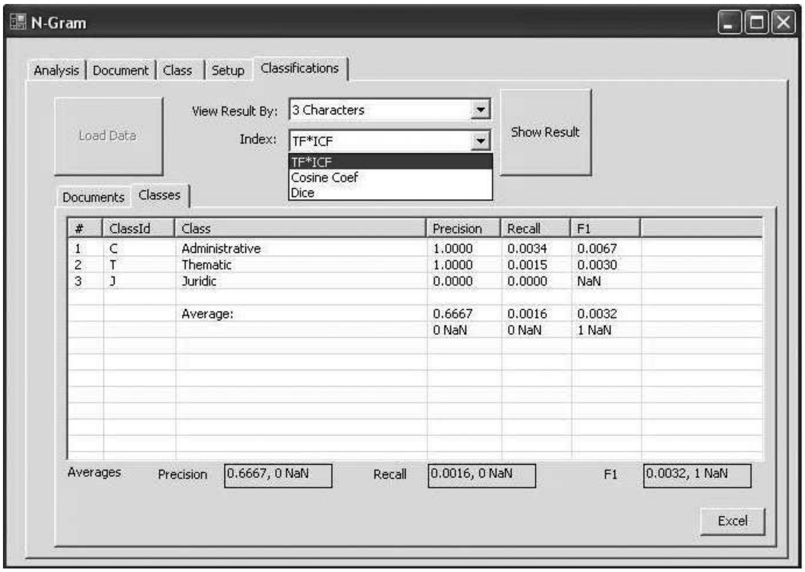

Figure 3.
Classifications result

*6.6 Level 1: TF*ICF*

The results are shown in Table III.

No.	ID	Class	Precision	Recall	F_1
1	C	Administrative	1.0000	0.0034	0.0067
2	T	Thematic	1.0000	0.0015	0.0030
3	J	Juridic	0.0000	0.0000	NaN
		Average:	0.6667	0.0016	0.0032
			0 NaN	0 NaN	1 NaN

Table III.
Level 1, TF*ICF classification

6.6.1 Cosine coefficient. The results are shown in Table IV.

No.	ID	Class	Precision	Recall	F_1
1	C	Administrative	0.9980	0.5753	0.7299
2	T	Thematic	0.9985	0.2475	0.3966
3	J	Juridic	0.9979	0.1784	0.3027
		Average:	0.9981	0.3337	0.4764
			0 NaN	0 NaN	0 NaN

Table IV.
Level 1, cosine coefficient classification

6.6.2 Dice. The results are shown in Table V.

No.	ID	Class	Precision	Recall	F_1
1	C	Administrative	0.0000	0.0000	NaN
2	T	Thematic	0.9985	1.0000	0.9992
3	J	Juridic	0.0000	0.0000	NaN
		Average:	0.3328	0.3333	0.3331
			0 NaN	0 NaN	2 NaN

Table V. Level 1, Dice classification

*6.7 Level 2: TF*ICF*

The results are shown in Table VI.

No.	ClassId	Class	Precision	Recall	F_1
1	C107	تعاونية موظفي الدولة	1.0000	1.0000	1.0000
2	C104	مشيخة عقل الطائفة الدرزية	1.0000	0.5000	0.6667
6	……	……	….	….	….
7	…….	……	….	….	….
420	C112	وزارة السياحة	0.0000	0.0000	NaN
421	C110	مجلس تنفيذ المشاريع الكبرى لمدينة بيروت	0.0000	NaN	NaN
422	C109	المديرية العامة للصناعة	0.3333	1.0000	0.5000
		Average:	0.3677	0.3107	0.2591
			0 NaN	140 NaN	215 NaN

Table VI. Level 2, TF*ICF classification

6.7.1 Cosine coefficient. The results are shown in Table VII.

No.	ClassId	Class	Precision	Recall	F_1
1	C107	تعاونية موظفي الدولة	0.5714	1.0000	0.7273
2	C104	مشيخة عقل الطائفة الدرزية	0.3333	1.0000	0.5000
3	……	……	….	….	….
4	…….	……	….	….	….
420	C112	وزارة السياحة	1.0000	0.0707	0.1321
421	C110	مجلس تنفيذ المشاريع الكبرى لمدينة بيروت	0.0000	NaN	NaN
422	C109	المديرية العامة للصناعة	0.5000	1.0000	0.6667
		Average:	0.4603	0.3257	0.3099
			0 NaN	140 NaN	174 NaN

Table VII. Level 2, cosine coefficient classification

6.7.2 Dice. The results are shown in Table VIII.

6.8 Discussion

We remark that the cosine coefficient measure in the two levels has given us the best results between the three measures used. In Level 1, the average F_1 in case of using cosine coefficient as similarity method is: 0.4764, and in Level 2 it is: 0.3099.

Table VIII.
Level 2, Dice classification

No.	ClassId	Class	Precision	Recall	F_1
1	C107	تعاونية موظفي الدولة	0.0000	0.0000	NaN
2	C104	مشيخة عقل الطائفة الدرزية	0.0000	0.0000	NaN
3	. . . . . .	. . ...	. . . .	. . ..	. . . .
4	. . . .	. . ...	. . . .	. . ..	. . ..
420	C112	وزارة السياحة	1.0000	0.0101	0.0200
421	C110	مجلس تنفيذ المشاريع الكبرى لمدينة بيروت	0.0000	NaN	NaN
422	C109	المديرية العامة للصناعة	0.0000	0.0000	NaN
		Average:	0.0283 0 NaN	0.0099 140 NaN	0.0083 403 NaN

Then it is the best between the three measures but it still insufficient. In the results above you will see the term NaN which means Not a Number, it means that we have in the denominator a zero, then the number is not defined.

In Level 1, we remark that the precision and recall for juridical class using TF*ICF and Dice coefficient is zero, that is because no correct classes or categories are found. We can explain by the expanding of juridical documents it means a juridical document can be considered as in administrative or thematic class.

In Level 2, the waste case was that of Dice coefficient in which contains 403 NaN may be because the Dice coefficient uses the intersection between the document and class divided by the sum of document and class. Then may be the ration very low for a document that belongs to a certain class.

XU, FRASER, WEISCHEDEL, 2002

7. Conclusions and future work

Arabic is one of the most widely used languages in the world, yet there are relatively few studies on the retrieval and classification of Arabic documents.

Laila KHREISAT, 2006

This paper presented the results of classifying Arabic text documents using the N-gram frequency statistics technique employing three similarity measures: TF*ICF, cosine coefficient, and Dice's measure of similarity.

Results showed that N-gram text classification using the cosine coefficient measure outperforms classification using the Dice's measure and TF*ICF weight.

This work evaluated a number of similarity measures for the classification of Arabic documents, using the Lebanese parliament documents and especially the Lebanese official journal documents Arabic corpus as the test bed.

We have proposed a segmentation method (N-gram) applied on Arabic documents, and our goal is to find the convenient similarity measure that gives us the powerful results when applied to Lebanese official journal documents.

N-gram method is good, but still insufficient for the classification of Arabic documents, and then we have to look at the future of a new approach like distributional or symbolic approach, and take in consideration that stemming can be important (Larkey and Connell, 2001; Larkey *et al.*, 2002; Goweder *et al.*, 2004) in order to increase the effectiveness.

Note

1. Wikipedia Encyclopedia. Available at: http://en.wikipedia.org/

References

Abdelali, A. (2004), *Improving Arabic Information Retrieval Using Local Variations in Modern Standard Arabic*, New Mexico Institute of Mining and Technology, New Mexico, NM.

Goweder, A., Poesio, M. and de Roeck, A. (2004), "Broken plural detection for Arabic information retrieval", *SIGIR'04*, Sheffield, 25-29 July.

Khoja, S. and Garside, R. (1999), "Stemming Arabic text", Computer Science Department, Lancaster University, Lancaster, available at: www.comp.lancs.ac.uk/computing/users / khoja/stemmer.ps

Larkey, L.S. and Connell, M.E. (2001), "Arabic information retrieval at UMass in TREC-10", *TREC 2001*, NIST, Gaithersburg, MD.

Larkey, L., Ballesteros, L. and Connell, M.E. (2002), "Improving stemming for Arabic information retrieval: light stemming and co-occurrence analysis", paper presented at SIGIR'02, Tampere, FL, 11-15 August, pp. 275-82.

Lavrenko, V. (2005), "Center for intelligent information retrieval", Hopkings IR Workshop, University of Massachusetts, Amherst, MA.

Mayfield, J., McNamee, P., Costello, C., Piatko, C. and Banerjee, A. (2002), "JHU/APL at TREC 2001: experiments in filtering and in Arabic, video, and web retrieval", in Voorhees, E. and Harman, D. (Eds), *Proceedings of the Tenth TextREtrieval Conference (TREC 2001), Gaithersburg, Maryland, MD*, July.

Corresponding author
Majed Sanan can be contacted at: Sinane80@hotmail.com