World of Computer Science and Information Technology Journal (WCSIT)
ISSN: 2221-0741
Vol. 2, No. 7, 126-129, 2013

A Comparative Study of Machine Learning Techniques in Classifying Full-Text Arabic Documents versus Summarized Documents

Dr. Khalil Al-Hindi, Eman Al-Thwaib
Computer Information Systems Department
University of Jordan
Amman, Jordan

Abstract- Text classification (TC) can be described as the act of assigning text documents to predefined classes or categories. Its necessity comes from the large amount of electronic documents on the web. The classification accuracy is affected by the content of documents and the classification technique being used.

Automatic text summarization is based on identifying the set of sentences that are most important for the overall understanding of document(s). The need for text summarization comes from the large amount of electronic documents and the need for saving processing time.

In this research, an automatic text summarizer has been used to summarize documents. Two classification methods have been used to classify Arabic documents before and after applying the summarization, then the classification accuracy of classifying the full documents and summarized documents have been compared. Classification accuracy resulted from classifying full documents is close to that resulted from classifying summarized documents. Nevertheless, memory space required and run time for classifying summarized documents are less than the memory and time needed for classifying full documents.

Keywords- Text Classification; Text Summarization; Naïve Bayes; k-Nearest Neighbors.

I. INTRODUCTION

With the amount of text files on Internet increases exponentially each day, the volume of information available online continues to expand. Text Classification (TC), as the assignment of text files to one or more predefined classes based on information contained from text files, is an important component in information management tasks. Automatic TC came to help human deal with the enormous amounts of data on web.

The motivation for feature reduction studies is the huge number of features or terms representing documents. If these terms can be reduced without affecting the value or content of documents, then memory space and classification processing time will be saved.

In our study, we study the effect of using the text summarization on classification accuracy. Summarizing the documents result in shortened form of these documents, so that the number of features or terms, that represent documents, will be reduced.

This paper is organized as follows. Section 2, briefly describe related works in TC field. Section 3 represents the core of our study/work and the implementation of two classification methods, Naïve Bayes (NB) and k-Nearest Neighbor (kNN), the summarizer being used is described. Experiments and results are discussed in section 4. In section 5, we present the conclusion of our work and experiments.

II. RELATED WORKS

Arabic is the mother language of more than 300 million people [1]. Unlike Latin-based alphabets, the orientation of writing in Arabic is from right to left; the Arabic alphabet consists of 28 letters. Arabic words have two genders; feminine and masculine, three number cases; 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.

Many methods are being used for TC such as NB, kNN, Support Vector Machine (SVM) [2], Neural Networks [3], N-gram [4], etc.

Reference [5] has used a corpus of 1500 Arabic web documents that are pre-classified into five classes (health, business, culture and art, science, and sport), 300 documents for each class. Documents were tokenized into words, stop words were removed, then the remaining terms/words were stemmed to their roots. NB classifier computes a posteriori probabilities of classes, using estimates obtained from a training set of labeled documents. When an unlabeled document is presented, the a posteriori probability is computed for each class using the Bayes theorem. Finally, the unlabeled document is assigned to the class which has the largest a posteriori probability.

Reference [6] applied kNN algorithm on a data set of 621 Arabic text documents. The documents were preprocessed by the system, the stop words were removed, a light stemmer was applied on the remaining tokens, and keywords were extracted. Normalized TF×IDF weighting scheme have been used to give those keywords weights. Data set was transformed into the Vector Space Model (VSM) [7], then vectors were split into two sets, training and testing sets. The system classifies a test document represented as a vector in the space model by comparing it to all training documents using the cosine similarity measure. k neighbors (of training documents) that have the highest similarity were taken into account in making decision for classifying the test document.

Reference [8] has used the NB algorithm to develop a spam email filter; they used pre-classified emails as training data. These emails have been used to train the filter, so it is able to decide whether an email is a spam email or not. The spam email filter has two stages, training stage and testing stage.

Reference [9] applied both NB and kNN classifiers on Arabic documents. In the Bayesian analysis, the new document X (to be classified) is classified based on the higher posterior probability.

III. THE PROPOSED APPROACH

We propose the following model for studying the classification based on the text summarization:

1. Documents are classified using a TC technique, so that the class of each document is predicted.

2. The same documents pass a text summarizer, the summaries resulted are classified, so a class for each document is predicted.

Sakhr Summarizer:

Automatic text summarization is the process in which a computer takes a text document(s) as input and produces a summary of that document(s) as an output. Many commercial applications are available such as Microsoft (MS) summarizer, Newsinessence summarizer, and Sakhr summarizer [10].

Reference [10] engine identifies the most relevant (to the topic or subject of document) sentences within a text and displays them as the text summary. The summarizer makes it easy to scan just the important sentences within a document, so that time needed to read and process documents manually is reduced. Key words extractor and spelling corrector are used. Finally the sentences forming the summary are highlighted.

Reference [10] engine employs a spelling corrector to automatically correct the input Arabic text from mistakes. Names are being treated as key words; in [10], proper names database contains 255,000 entries with different types, and it grows continuously

A common problem in TC is the high number of terms or features in document(s) to be classified, where d= {w1,w2,…,wi}. This problem can be solved by selecting the most important terms.

We have used the text summarization as terms selection method. The summarizer of [10] has been used to summarize our corpus.

Data Preprocessing:

The data set/corpus that we used, consists of 1000 Arabic text documents. It is a subset of 60913-document corpus collected from many newspapers and other web sites. The 1000 documents were pre-classified to five different classes (Economy, Politics, Religion, Science, and Sport), 200 documents for each class.

The 1000 Arabic documents have been preprocessed before being used, each document have been tokenized, i.e. split it into tokens according to the white space position. Tokens that less than 3 letters were removed, then:

1. Punctuations (such as ! ؛ , . ؟), symbols (such as < > }]), and digits have been removed. The comma ” ,” has a special case, because it appears sometimes connected to a word (without a space in between). Our preprocessor searches the beginning and end of tokens for a comma and removes it.

2. Non-Arabic words have been removed.

3. Stop words (such as عن ,لكن ,في) have been removed.

4.Remaining terms have been normalized, i.e., Letters “ ء”, “ آ ” , “ أ ” , “ ؤ ”, “ ئ ”, and “ ئ” have been replaced with “ا”, letter “ى ” replaced with “ي”, and the letter “ة ” replaced with “ه ”

Naïve Bayes and k-Nearest Neighbor implementation:

We will classify documents before summarization (full documents) and after summarization. We hope that the summarized documents will contain the most relevant words and hence give better classification results. We will compare the results using two machine learning methods: NB classifier and kNN. Two phases are implemented, training and testing. 10-fold cross validation is used to split data set into training and testing data, i.e., data is divided into 10 equal parts. One part is used for testing and the remaining nine parts for training the classifier. This operation is

repeated 10 times with different testing data part each time, finally the results are averaged.

A Bayes classifier is a simple probabilistic classifier based on applying Bayes theorem with independence assumption as in:

$$p(v|d) = \frac{p(d|v)p(v)}{p(d)} \quad (1)$$

P(*v*|*d*) is called posterior probability of v given d. In TC, we consider a set of classes *v1, v2, . . . , vk* and a set of text documents *d1, d2, . . . , dm* each with known class. Document d consists of words sequence *w1, w2, . . . ,wn*. We need to find Maximum A Posteriori (MAP) class given this document d.

kNN is a good example of instance-based classifiers. In order to decide whether a document d belongs to the class c, kNN classifier checks whether the k training documents (that are most similar to d) belong to c. If the answer is positive for a large proportion of them, a positive decision is made; otherwise, the decision is negative. We have used the VSM to represent the documents, each vector represents one document and it has the weights of tokens that result from the preprocessing of that document. The weighting scheme used is the Term Frequency (TF), which represents the number of times the term repeats in one document. We have used k=30, i.e. 30 neighbors are taken into account. The cosine similarity measure (the cosine of the angle between vectors) is used to calculate the similarities between documents.

To classify the document x, the similarity between the test document and every document in the training set is calculated; here we use the cosine similarity measure. It measures the cosine of the angle between the test document vector and a training document vector as follows:

$$\text{Sim (d,x)} = \frac{\sum_{i=1}^{t}(wdi \times wxi)}{\sqrt{\sum_{i=1}^{t} wdi^2 \times \sum_{i=1}^{t} wxi^2}} \quad (2)$$

Where *wdi* is the weight of term i in document d. In the carpetbag, we are interested in only common terms between the test and training document, but in the denominator, all terms of the document will be taken into account. After calculating similarities, 30 nearest neighbors to the test document are determined (k-list) then a score is given for each class by counting the number of documents that appear in the k-list and belong to that class, Classes scores are sorted in descending order and the document x is assigned to the class with the highest score. This is repeated for all test documents then the classification accuracy is calculated.

IV. RESULTS

The performance of the NB and kNN classifiers (in classifying the full and the summarized documents) is measured with respect to the accuracy. Accuracy can be measured by (3):

$$\text{Accuracy} = \frac{number\ of\ correctly\ classified\ documents}{total\ number\ of\ testing\ documents} \times 100\% \quad (3)$$

Also the time needed for classification and memory space requirement are taken into account.

Classification using the NB results in shorter running time for the summarized documents; 20 minutes and 35 seconds for the full corpus, 6 minutes and 43 seconds for the summarized documents. Experiments were done on hardware with 2.13 GHz processor and 3GB of RAM. The shorter run time is justified by the less number of features or terms.

The main difference is between the space required to store the probabilities of each word in each class P(wk |vj) in different 10 experiments for the same corpus or data set. The memory space required to store data for full-documents corpus is about 8MB on average, while the memory space required to store data of summarized-documents corpus is about 4MB on average. That means that text summarization technique used helps saving memory requirement.

The classification accuracy (the percentage of correctly classified documents among all training documents) for classifying full documents by NB is 97.1% and 96.5% for classifying summarized documents, as average for categories results shown in fig. 1:

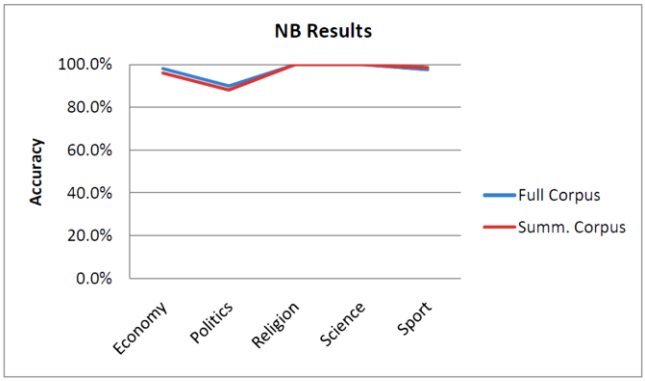

Figure 1. Classification results using NB

We succeed to use summarization for term selection and overcoming kNN drawbacks. Term reduction results in smaller memory requirement (about 4MB size for full corpus inverted file and 2MB for summarized corpus inverted file). Less time will be needed for classification; 1 hour and 59 minutes and 48 seconds for classifying the full-document corpus, and 14 minutes and 27 seconds for classifying the summarized documents using the same machine described before.

Therefore, the accuracy is 93.1% for full documents and 92.5% for summarized documents, as average for categories results shown in fig. 2:

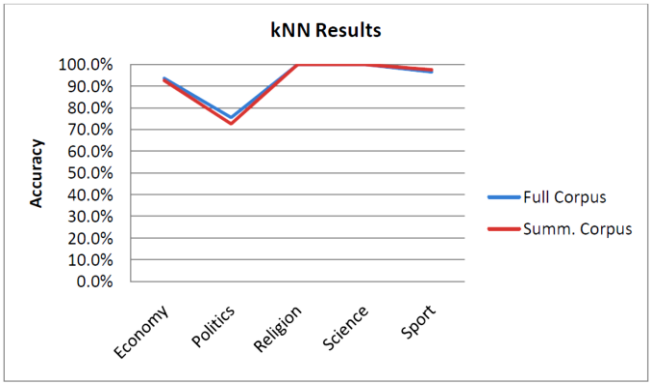

Figure 2. Classification results using kNN

Although the classification results are close to each other, but we found that it is feasible to use the summarizer to save time and memory.

V. CONCLUSION

Text documents are continuously increasing every day, so that long time will be spent to deal with all that documents. Automatic TC has come as a solution for that problem. Although it is not 100% accurate but it saves the time needed to read documents and still gives results that are close to those given by human (depending on the classification method being used).

In this study, we have proposed a way to reduce the number of terms that represent the document in classification process. Automatic text summarization is proposed to solve the problem of high dimensions in the feature space, i.e. to reduce the number of features. We have applied two TC methods in our experiments, the NB and the kNN. We have classified the full documents then classified the summaries of those documents using the same classifiers.

Although the results of classification before and after summarization were close in accuracy, but we believe that the time and the memory space that have been saved does worth it.

REFERENCES

[1] Al-Harbi S., Almuhareb A., Al-Thubaity A., Khorsheed M. S., and Al-Rajeh A. (2008), "Automatic Arabic text classification". JADT: Journees internationales. Pages 77-83.

[2] Mesleh Abdelwadood. Moh'd. (2007), "Support vector machines based Arabic language text classification system: feature selection comparative study". Advances in computer and information sciences and engineering. Pages 228-233.

[3] Goyal Ram Dayal (2007), "Knowledge based Neural Network for text classification". IEEE international conference on granular computing. d'Analyse statistique des Donnees Textuelles. Pages 542-547.

[4] Khreisat Laila (2006), "Arabic text classification using N-Gram frequency statistics, a comparative study". Proceedings of the international conference on data mining (DMIN2006). Las Vegas, USA. Pages 78-82.

[5] El-Kourdi Mohamed, Bensaid Amine, and Rachidi Tajje-eddine (2004), "Automatic Arabic documents categorization based on the Naïve Bayes algorithm". In proceedings of the workshop on computational approaches to Arabic script-based languages (COLING-2004), University of Geneva, Geneva, Switzerland. Pages 51-58.

[6] Al-Shalabi Riyad, Kanaan Ghassan, Gharaibeh Manaf H. (2006), "Arabic text categorization using kNN algorithm". Proceedings of the 4th international multiconference on computer science and information technology (CSIT 2006).Volume 4. Amman, Jordan.

[7] Salton G., Wong A., and Yang C. S. (1975), "A vector space model for automatic indexing". Communications of the ACM. Volume 18. Number 11. Pages 613-620.

[8] Zhang Haiyi, Li Di (2007), "Naïve Bayes text classifier". IEEE international conference on granular computing. Pages 708-711.

[9] Bawaneh Mohammed J., Alkoffash Mahmud S., and Al Rabea Adnan I. (2008), "Arabic text classification using K-NN and Naïve Bayes". Journal of computer science 4 (7), pages 600-605.

[10] Sakhr company website: http//:www.sakhr.com .last visit in June, 2010.