Arabic Text Classification using K-Nearest Neighbour Algorithm

Roiss Alhutaish and Nazlia Omar
Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, Malaysia

Abstract: *Many algorithms have been implemented to the problem of Automatic Text Categorization (ATC). Most of the work in this area has been carried out on English texts, with only a few researchers addressing Arabic texts. We have investigated the use of the K-Nearest Neighbour (K-NN) classifier, with an I_{new}, cosine, jaccard and dice similarities, in order to enhance Arabic ATC. We represent the dataset as un-stemmed and stemmed data; with the use of TREC-2002, in order to remove prefixes and suffixes. However, for statistical text representation, Bag-Of-Words (BOW) and character-level 3 (3-Gram) were used. In order to, reduce the dimensionality of feature space; we used several feature selection methods. Experiments conducted with Arabic text showed that the K-NN classifier, with the new method similarity I_{new} 92.6% Macro-F1, had better performance than the K-NN classifier with cosine, jaccard and dice similarities. Chi-square feature selection, with representation by BOW, led to the best performance over other feature selection methods using BOW and 3-Gram.*

Keywords: *ATC, K-NN, similarity measures, feature selection methods.*

Received May 3, 2012; accepted February 13, 2014; published online April 23, 2014

1. Introduction

With the exponential growth in the availability of online information and continuously increasing documents in digital form, there is a need to classify documents so that we can access their sources. Many machine learning algorithms have been applied to the text categorization task, which is considered to be one of many information management tasks. During the early eighties, hardware had a limited capacity. Therefore, most work was aimed at researching for new methods to store, represent and retrieve relevant information from a small number of documents. However, the 90's was considered to be the starting point for research interested in the meaning of text, when the internet became a freely-accessible facility to everybody, anywhere and anytime [17].

Text categorization (or classification) is one of the most important problems in machine learning and data mining, due to the huge amount of information on the internet increasing the availability of electronic documents and information libraries. The idea of text classification assigns one document to one or more categories, based on its contents.

There are mainly two classification approaches to enhance the organizational task of digital documents. First is the supervised approach, which is commonly used where a pre-defined category is labelled and assigned to a document based on its contents. Text categorization systems classify new documents into one label that is determined by predefined categories. Second is the unsupervised approach, which is also applied where there is no need for human intervention or labelled documents at any point in the whole process [14]. There are many supervised learning algorithms that have been applied to the area of text classification, using pre-classified training document sets. Those algorithms, that used classification, include K-Nearest Neighbour (K-NN) classifier, Naïve Bayes (NB), decision trees, rocchio's algorithm, Support Vector Machines (SVM) and Neural Networks [8, 11, 12, 16, 17].

2. Related Work

Many algorithms have been applied for Automatic Text Categorization (ATC). Most studies have been devoted to English and other Latin languages. However, very few researches have been carried out on Arabic text. For example, Al-Shalabi *et al.* [4] proposed project for Arabic text classification using K-NN, based on a similarity score; El-Halees [10] implemented a maximum entropy based classifier, to classify Arabic documents; Duwairi [9] proposed a paper, which methods of Arabic text had three classification, namely: K-NN, NB and distance based; Al-Kabi and Al-Sinjilawi [2] proposed a comparative study of classifying Arabic text between naïve bayesian and euclidean; when they used four different methods of coefficients of the vector space model: Cosine, dice, jaccard and inner product; Bawaneh *et al.* [5] proposed a paper to compare between two classifiers, namely K-NN and NB; Al-Harbi *et al.* [1] evaluated the performance of the C5.0 decision tree and the SVM classification algorithm; Khreisat [13] proposed a paper for Arabic text document

classification using N-Gram; Thabtah *et al.* [18] investigated the term weighting approaches, when applying these three methods of similarity: Cosine, jaccard and the based K-NN algorithm and Al-Salemi and AbAziz [3] evaluated three models of Bayesian learning, in order to enhance Arabic ATC, namely simple NB, Multinomial Naïve Bayes (MNB) and Multi-variant Bernoulli Naïve Bayes (MBNB).

Arabic consists of 28 letters; three of which are long vowels and the rest are consonants. It is written from right to left, has a very complex morphology and the majority of words have a tri-letter root. In addition, the characters of the Arabic alphabet have only one form; unlike English, which contains two forms of the same letter i.e., capital and lowercase.

3. Methodology

Text categorization incorporates a number of stages. Figure 1 shows a general overview of the architecture of a text categorization system. These stages are text pre-processing, feature selection and classifier.

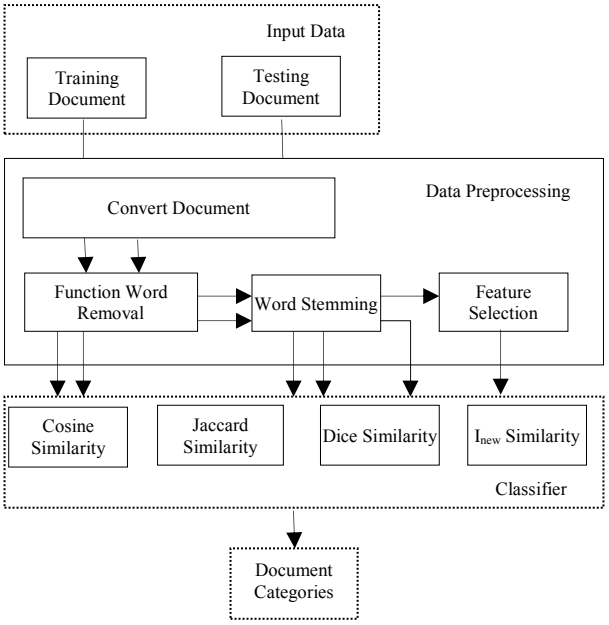

Figure 1. The architecture of text categorization.

3.1. Text Pre-processing

The first step of Arabic text reprocessing, like any ACT system, is pre-processing of the text. There are two types of features, namely un-stemmed data and stemmed data. Un-stemmed data are the words extracted from the documents as they are [11, 12]. For Arabic text, function word removal: Removes punctuation marks, diacritics, non-letters, stop words, and the elimination of words with a length of less than three [13, 15]. Stemmed data are the stems of the extracted words [12]. This study used a Light stemmer, also known as TREC-2002 Light Stemmer, which removes the most frequent prefixes and suffixes [6]. This study used both stemmed and un-stemmed data.

3.2. Feature Selection

Feature selection is one of the most important tasks that improve the performance of text classification by removing the features that are considered irrelevant for classification, in order to, reduce the dimensionality of the dataset.

Let us suppose that c_i is a category in category set $\{c_1, c_2, ..., c_{|C|}\}$; t is a term that belongs to one or more documents in a training set; N is the total number of training documents; A is the number of documents in class c_i that contain t, B is the number of documents that contain the term t in other classes; C is the number of documents in class c_i that does not contain the term t; and D is the number of documents that does not contain the term t in other classes.

$$CHI(t,c_i) = \frac{(N \times (AB+CD))}{(A+C)(B+D)(A+B)(C+D)} \quad (1)$$

$$GSS(t,c_i) = P(t,c_i) \times P(\bar{t},\bar{c_i}) - P(\bar{t},\bar{c_i}) \times P(\bar{t},\bar{c_i}) \approx \frac{AD-CB}{N^2} \quad (2)$$

$$OR(t,c_i) = \frac{P(t \mid c_i) \times \{P(t \mid \bar{c_i})\}}{\{P(t \mid c_i)\} \times P(t \mid \bar{c_i})} \approx \frac{AD}{CB} \quad (3)$$

$$MI \approx \log_2 \frac{A \times N}{(A+C) \times (A+B)} \quad (4)$$

Two different measures can be computed for each FS method: Max score or average score. In this paper, we used Max score.

$$FS_{(ave-score)}(t) = \sum_{i=1}^{m} P(c_i) \times FS(t,c_i) \quad (5)$$

$$FS_{maxscore}(t) = Max_{i=1,2,...m} \{FS(t,c_i)\} \quad (6)$$

In Equation 6, $FS_{max-score}$ returns the accordant category that t belongs to.

3.3. Classifiers

The K-NN algorithm, which was introduced by Dasarathy in [7], is one of most famous algorithms in the field of text classification, which gives good accurate results and is easy to understand. However, it is a lazy learning algorithm that depends only on statistics. Another disadvantage of K-NN is choosing the best value of K. In general, text classification using the K-NN classifier can be summarized as follows: Assign each training document with a predefined label, compute the similarity between a test document and every training document based on the value of K, sort the documents into descending order of their similarity to the test document in which the highest similar values are chosen, assign the test document to a category that has the highest score of similarity. In our paper, we investigated the use of the K-NN classifier with a cosine, jaccard, dice and new method (I_{new}) similarities (see of Equations 7, 8, 9 and 11, respectively), in order to enhance Arabic ATC.

3.4. Similarity Measuring

Cosine, jaccard and dice, are similarity functions that are commonly used with the K-NN classifier. Assuming that D_i and D_j are vectors that represent testing and training documents respectively, we can calculate the similarity between D_i and D_j using the following formulas:

$$Sim_{cosine}(D_i, D_j) = \frac{\sum_{k=1}^{m}(W_{ik} \times W_{jk})}{\sqrt{\sum_{k=1}^{m}(W_{ik})^2 \times \sum_{k=1}^{m}(W_{jk})^2}} \quad (7)$$

$$Sim_{jaccard}(D_i, D_j) = \frac{\sum_{k=1}^{m}(W_{ik} \times W_{jk})}{\sum_{k=1}^{m}(W_{ik})^2 + \sum_{k=1}^{m}(W_{jk})^2 - \sum_{k=1}^{m}(W_{ik} \times W_{jk})} \quad (8)$$

$$Sim_{dice}(D_i, D_j) = \frac{2 \times \sum_{k=1}^{m}(W_{ik} \times W_{jk})}{\sum_{k=1}^{m}(W_{ik})^2 + \sum_{k=1}^{m}(W_{jk})^2} \quad (9)$$

Where, D_i is the test document, D_j is the training document, W_{ik} corresponds to the weight of the k^{th} element of the term vector D_i and W_{ik} is the Weight of the k^{th} element of the term vector D_j.

I_{new} is a new method, proposed by Duwairi and Al-Zubaidi [8]. This method is based on the equation used to calculate the ID3. Let 's' be the set of tuples in the database. Let 'm' be the set of distinct classes C_i (for $i=1,\ldots, m$). Let 'S_i' be the number of tuples in class C_i.

$$I_{new}(S_1, S_2, \ldots S_n) = -\sum P_i log_2(P_i), i = 1, 2, \ldots m \quad (10)$$

Where, p_i is the probability that an arbitrary tuple belongs to class C_i and is estimated by $(\frac{S_i}{S})$. Let us suppose that 'm' is the number of distinct terms in the document that represents the category. Let 'S' be the total number of occurrences for the terms in the training document. Let 'S_i' be the frequency of terms in the test document that are shared with the training document.

$$I_{new}(S_1, S_2, \ldots S_n) = -\sum_{i=1}^{m} \frac{S_i}{S} log_2(\frac{S_i}{S}) \quad (11)$$

3.5. Performance Measures

Many measures are used to evaluate various aspects of text processing and information retrieval systems. The performance of such a system, which is designed to classify documents to their categories, is often gauged in terms of precision, recall and macro-average. Let True Positives (TP) be the number of documents that are classified as relevant, judged by the human and the classifier TP, False Negatives (FN) be the number of documents that are classified as relevant by judgment of the human and irrelevant by judgment of the Classifier FN, False Positives (FP) be the number of documents that are classified as irrelevant by judgment of the human and relevant by judgment of the classifier FP and True Negatives (TN) be the number of documents that are classified as irrelevant by judgment of the human and the classifier TN. Recall and precision are defined respectively as:

- *Precision*: Measures that have a high ability to retrieve documents that are judged by the user as being relevant.

$$Precision = \frac{TP}{TP + FP} \quad (12)$$

- *Recall*: Measures that have a high ability of the search to find all of the relevant items in the corpus.

$$Rcall = \frac{TP}{TP + FN} \quad (13)$$

- *Macro-averaged*: F-measurement combines *Precision* and *Recall* and is defined as follows.

4. Experimental Results

The dataset used in our system consists of 3,172 documents, distributed into four categories: Arts, economic, politics and sport. The dataset was collected from the website and divided into a 1,732 document training set and a 1,440 document testing set. Table 1 shows a breakdown or the respective dataset categories. This dataset was collected by [3].

Table 1. The categories and their training and test sets.

	Art	Politics	Economic	Sport
Training Set	414	430	543	345
Test Set	360	360	360	360

4.1. Experiments

Four experiments were conducted using three types of data. These experiments were a measurement of performance of I_{new} similarity against the cosine, jaccard and dice similarities. The three phases dealt with were: Unstemmed data (EXP1), stemmed data (EXP2), four methods of feature selection (i.e., Chi-Square Statistic (CHI), GSS Coefficient (GSS), Mutual Information (MI) and Odds Ratio (OR)) stemmed data (EXP3) and the four experiments (EXP4), which is a time measurement to previous experiments. In each phase, we represented datasets with Bag-Of-Word (BOW) by using simple words as features and N-Gram by using sequence characters (character level N-Gram) with the length n, where we used Tri-Gram (3-Gram).

4.2. Results

We investigated the performance of the K-NN classifier with I_{new}, cosine, jaccard and dice similarities, according to unstemmed data, stemmed data and each FS method (i.e., CHI, GSS, MI and OR); by selecting a variable number of the top most frequent terms in each feature set (3-Gram and BOW).

Figure 2 shows that the K-NN classifier with I_{new} similarity achieved the best performance when applied to EXP1 with 3-Gram. The best performance of I_{new} similarity, according to Macro-F1, was 80%. Dice and jaccard similarities obtained 79% Macro-F1 and cosine obtained 77.91% Macro-F1. However, cosine similarity achieved the highest value of macro-F1 when EXP1 was represented by BOW. It obtained 86.09%, while I_{new}, jaccard and dice, similarities obtained 84.43%, 83.75% and 83.5% macro-F1, respectively.

Figure 3 shows that the K-NN classifier with cosine obtained the highest value of macro-F1, when EXP2 was represented by 3-Gram. It obtained 87.55% macro-F1, jaccard obtained 84.28%, 84.02% for dice and 81.73% for I_{new}. Furthermore, it achieved the best performance when EXP2 was represented by BOW. They obtained 88.2%, 87%, 86.34% and 86.02% macro-F1 scores for cosine, dice, jaccard and I_{new} similarities, respectively. Based on the results obtained from the first and second experiments, we saw that the results of the second experiment were better because stemmed data was used in the second experiment. The stemmed data increased term sharing between training documents and testing documents, as well as an increase in term frequency.

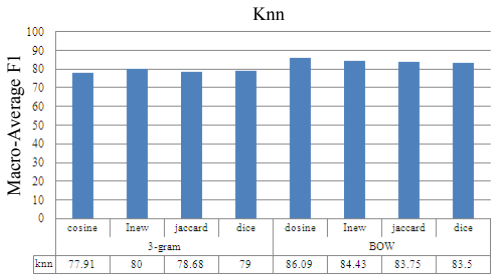

Figure 2. Macro-F1 values of cosine, jaccard, dice and I_{new} similarityclassifiers, without stemmer based 3-Gram and BOW.

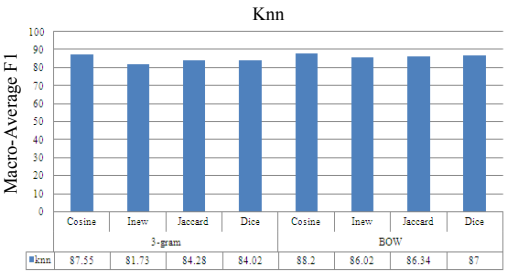

Figure 3. Macro-F1 values of cosine, jaccard, dice and I_{new} similarity classifiers, with stemmer based 3-Gram and BOW.

The third experiment (EXP3) consisted of two parts. Figure 4 considers the first part of EXP3. This part was represented by 3-Gram and used four feature selection methods, namely CHI, GSS, OR and MI.

I_{new} similarity obtained the best performance, which were represented by 3-Gram. It obtained 91% macro-F1 when more than one Nearest Neighbour (NN) was used. This result was obtained when MI was implemented with I_{new}, cosine, jaccard and dice similarities. Cosine similarity obtained the best performance with GSS. It obtained 88.5% macro-F1. Meanwhile, jaccard and dice similarities obtained 89% macro-F1 when GSS was used.

We observed that the performance of I_{new} similarity was better than cosine, jaccard and dice similarities, when light stemmer with feature selection methods was used. I_{new} similarity gave better results when term sharing is increased. Based on the results obtained from the second and EXP3, which were represented by 3-Gram, we observed that the results of the EXP3 were better, because feature selection methods were used. Feature selection methods choose the features that are considered relevant for the category and thus, obtain better performance.

Figure 5 considers the second part of EXP3. This part was represented by BOW. This experiment used four feature selection methods, namely CHI, GSS, OR and MI.

I_{new} similarity obtained the best performance in this study when CHI was implemented with I_{new}, cosine, jaccard and dice similarities. It obtained 92.6% macro-F1 when more than one NN was used. Cosine similarity obtained the best performance with GSS. It obtained 88.8% macro-F1. Jaccard and dice similarities obtained 88.6% and 88.3% macro-F1, respectively; when CHI was used.

Based on the results shown in Figure 4, we observe that I_{new} similarity achieved a better performance than cosine, dice and jaccard similarities. It was superior in all stages. Furthermore, the results of BOW were better than the results of 3-Gram in all experiments. BOW is more meaningful to characterize documents to a set of tokens, because BOW provides tokens as natural words, as they appear in the document. 3-Gram generates many terms that occur only once in a category; frequent terms become quite rare in other documents. This drawback affects the overall performance.

Figure 4. Macro-F1 values of cosine, jaccard, dice and I_{new} similarity classifiers using a feature selection method based on 3-Gram.

Figure 5. Macro-F1 values of cosine, jaccard, dice and I_{new} similarity classifiers, using a feature selection method based on BOW.

The EXP4 evaluated the three previous experiments, by calculating the time required to find similarities between training and test documents. In Table 2, we observed that I_{new} similarity had a shorter execution time than jaccard, dice and cosine similarities, when executing the three previous experiments. This difference in time, which was spent between cosine, jaccard, dice similarities and I_{new} similarity, was because I_{new} similarity took only the shared terms between the test and training document; whereas cosine, jaccard and dice similarities took all possible terms.

Table 2. Time taken for the implementation of each experiment.

		I_{new}	Cosine	Jaccard	Dice
EXP1	3-Gram	92.234	196.625	204.406	194.109
	BOW	100.297	394.141	393.297	393.875
EXP2	3-Gram	55.828	127.563	127.188	127.235
	BOW	99.656	341.469	350.171	344.188
EXP3	3-Gram	50.384	120.113	120.172	121.006
	BOW	49.732	182.08	204.414	208.72

5. Conclusions

Many experiments have been run on the K-NN classifier, using the four similarities of cosine, jaccard, dice and I_{new}. The majority found that the I_{new} classifier was the best.

The process of classifying a new document using the I_{new} similarity, showed that less time was needed compared to cosine, jaccard and dice similarities in all experiments. The computation of the classification using the I_{new}, K-NN classifier dealt only with the frequency of shared features between the test documents and the category representative with simple computational operation. Mean while, the computation of the classification using the cosine, jaccard and dice K-NN classifiers, dealt with the frequency of all data space features.

References

[1] Al-Harbi S., Almuhareb A., Al-Thubaity A., Khorsheed S., and Al-Rajeh A., "Automatic Arabic Text Classification," *in Proceedings of the 9th International Conference on the Statistical Analysis of Textual Data*, France, pp. 77-83, 2008.

[2] Al-Kabi M. and Al- Sinjilawi S., "A Comparative Study of the Efficiency of Different Measures to Classify Arabic Text," *University of Sharjah Journal of Pure and Applied Sciences*, vol. 4, no. 2, pp. 13-26, 2007.

[3] Al-Salemi B. and Ab Aziz M., "Statistical Bayesian Learning for Automatic Arabic Text Categorization," *the Journal of Computer Science*, vol. 7, no. 1, pp. 39-45, 2011.

[4] Al-Shalabi R., Kanaan G., and Gharaibeh M., "Arabic Text Categorization using K-NN Algorithm," *in Proceedings of the 4th International Multi-conference on Computer Science and Information Technology*, Amman, Jordan, pp. 1-9, 2006.

[5] Bawaneh J., Alkoffash S., and Al Rabea I., "Arabic Text Classification using K-NN and Naive Bayes," *Journal of Computer Science*, vol. 4, no. 7, pp. 600-605, 2008.

[6] Darwish K. and Oard W., "CLIR Experiments at Maryland for TREC-2002: Evidence Combination for Arabic-English Retrieval," *in Proceedings of Text Retrieval Conference*, Gaithersburg, USA, pp. 703-710, 2002.

[7] Dasarathy B., *Nearest Neighbour Norms: NN Pattern Classification Techniques*, *IEEE Computer Society Press*, California, USA, 1991.

[8] Duwairi R. and Al-Zubaidi R., "A Hierarchical K-NN Classifier for Textual Data," *the International Arab Journal of Information Technology*, vol. 8, no. 3, pp. 244-252, 2011.

[9] Duwairi R., "Arabic Text Categorization," *the International Arab Journal of Information Technology*, vol. 4, no. 2, pp. 125-131, 2007.

[10] El-Halees M., "Arabic Text Classification using Maximum Entropy," *the Islamic University Journal*, vol. 15, no. 1, pp. 157-167, 2007.

[11] He J., Tan A., and Tan C., "A Comparative Study on Chinese Text Categorization Methods," *in Proceedings of the International Workshop on Text and Web Mining*, Melbourne, Australia, pp. 24-35, 2000.

[12] Joachims T., "Text Categorization with SVM: Learning with Many Relevant Features," *in Proceedings of 10th European Conference on Machine Learning*, Chemnitz, Germany, pp. 137-142, 1998.

[13] Khreisat L., "A Machine Learning Approach for Arabic Text Classification using N-Gram Frequency Statistics," *the Journal of Informetrics*, vol. 3, no. 1, pp. 72-77, 2009.

[14] Ko Y., "Text Categorization using Unlabelled Data," *PhD Dissertation University Seoul*, Korea, 2003.

[15] Soudi A., Bosch A., Neumann G., *Arabic Computational Morphology*, Springer, 2007.

[16] Saleeb H., "Information Retrieval: A Framework for Recommending Text-Based Classification Algorithms," *PhD Doctor of Professional Studies*, *Pace University*, 2002.

[17] Sebastiani F., "Machine Learning in Automated Text Categorization," *ACM Computing Surveys,* vol. 34, no. 1, pp. 1-47, 2002.

[18] Thabtah F., Hadi W., and Al-shammare G., "Vsms With K-Nearest Neighbour To Categorise Arabic Text Data," *in Proceedings of the World Congress on Engineering and Computer Science*, San Francisco, USA, pp. 778-781, 2008.

Roiss Alhutaish is PhD student in UKM, Malaysia. He earned his MSc degree in 2011 in computer science from UKM, Malaysia. BSc degree in 2003 from Mu'tah University, Jordan. He worked as a lecture at Aden Community College from 2004-2005, Yemen. His research interests areon data mining and information retrieval.

Nazlia Omar is currently an Associate Professor at the School of Computer Science, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia. She holds her PhD degree in computer science from the University of Ulster, UK. Her main research interest is in the area of natural language processing, database and information systems.