AN EFFECTIVE ARABIC TEXT CLASSIFICATION APPROACH BASED ON KERNEL NAIVE BAYES CLASSIFIER

Raed Al-khurayji[1] and Ahmed Sameh[2]

[1,2]Faculty of Computer and Information Science, University Prince Sultan University, Riyadh, Saudi Arabia.

ABSTRACT

With growing texts of electronic documents used in many applications, a fast and accurate text classification method is very important. Arabic text classification is one of the most challenging topics. This is probably caused by the fact that Arabic words have unlimited variation in the meaning, in addition to the problems that are specific to Arabic language only. Many studies have been proved that Naive Bayes (NB) classifier is being relatively robust, easy to implement, fast, and accurate for many different fields such as text classification. However, non-linear classification and strong violations of the independence assumptions problems can lead to very poor performance of NB classifier. In this paper, first, we pre-process the Arabic documents to tokenize only the Arabic words. Second, we convert those words into vectors using term frequency and inverse document frequency (TF-IDF) technique. Third, we propose an efficient approach based on Kernel Naive Bayes (KNB) classifier to solve the non-linearity problem of Arabic text classification. Finally, experimental results and performance evaluation on our collected dataset of Arabic topic mining corpus are presented, showing the effectiveness of the proposed KNB classifier against other baseline classifiers.

KEYWORDS

Arabic Language, Text Classification, Machine Learning, Naïve Bayes Classifier, Kernel Estimation Function.

1. INTRODUCTION

Online or offline storage integrated with intelligent tools facilitated access to information stored on electronic documents in an effective manner. These electronic documents can be easily processed and also classified through these intelligent tools. According to [1], text classification (TC) is the activity of labelling the texts of natural language with a pre-specified set of thematic categories. Therefore, TC of electronic documents mainly refers to the categorization of the documents based on their contents into their relevant groups. High growth of text information in Internet and Big data environment resulted in billions of electronic documents to be created, edited and stored digitally [2].

Arabic language is one of the most widely spoken semantic, sharing many commonalities with other semantic languages in terms of vocabularies, vowels, morphologies and word orders [3, 4]. Arabic topic classification (ATC) is considered one of the most challenging research topics. This is probably caused by the fact that Arabic words have unlimited variation in the meaning, in addition to the problems that are specific to Arabic language only. Using Natural Language Processing (NLP) tools coupled with TC algorithms, it is possible to automatically identify the semantic content of electronic documents and group them according to their topics. Topic detection or classification process is achieved through the training a model with the corpus of

DOI : 10.5121/ijaia.2017.8601

every topic using the term of word vector. Arabic topic detection algorithm involves identifying and selecting the topics from the Arabic documents. This is achieved through various tools such as a Vector Space Model (VSM) [5]. In this context. It is worth mentioning that, single document summarizer is performed using a VSM where the sentences of Arabic document file are taken to generate an executive summary. So, automatic text summarization is an important tool in the Arabic topic detection [6].Actually, process mining of text is mainly performed based on a priori architectural design and the term of word vectors. Arabic topic identification and Arabic topic mining algorithms are two techniques used to identify and classify the text documents based on the specific topics in an online or offline manner [7].

In this paper, we introduce a new approach to classify the Arabic text using the kernel naive Bayes classifier. The proposed kernel Bayes classifier is more efficient than other baseline classifiers, including the traditional naive Bayes classifier which are used in the literature of Arabic topic classification task. At first, term frequency and inverse document frequency (TF-IDF) technique were used to generate the textual features of a corpus that contains Arabic text documents. The generated features are then used along with the kernel Bayes classifier to classify the testing set documents. Then, a number of baselineclassifiers are employed for the comparison purpose. These classifiers include Naïve Bayes (NB), Bayes Nets (BN), K-Nearest Neighbours (KNN), Decision Tree (J48), Support Vector Machine (SVM) and Hidden Markov Model (HMM). For implementation, an open source machine-learning Weka tool is used for data pre-processing, feature extraction and classification.

In next section, a literate review of related work is presented. Section 3 introduces the proposed methodology. Section 4 discusses the experiments and evaluation for the proposed. Finally,the conclusions are summarized in Section 5.

2. RELATED WORK

Text classification is becoming a very important task with the presence of huge text information. Online or offline electronic large scale documents in many different languages are increasing every day [5]. Arabic text documents are part of this growth and there is increasingly need to retrieve, filter and mine in many applications. In [8] developed a text classification system for Arabic language. This system compares the representation of document by N-grams (unigrams and bigrams) and single terms (bag of words) as a feature extraction method in the pre-processing step. Afterwards, TF-IDF is used to reduce dimensionality and KNN classifier is applied for classification of Arabic text. The experimental results showed that using unigrams and bigrams as representation of documents outperformed the use of bag of words in term of accuracy.

In [9], a statistical method called maximum entropy is used to classify Arabic news article. In [10], designed a multi-word term extraction method as a feature extraction of Arabic language. They used a hybrid method to extract multi-word terminology from Arabic corpus. From the respective of linguistic, they used some linguistic information to extract and filter the candidates of multiword terminology. In [11], SVM is used to classify the Arabic text and compared result with KNN classifier. Arabic Topic Identification Algorithm involves the identification and selection of topics from an Arabic document, and it was achieved through various tools including the Vector Space Model (VSM). Single document summarizer is performed using the Vector Space Model which takes Arabic document and the initial sentence of the file to generate an executive summary. The automatic text summarization presents to be an important tool in the Arabic Topic Detection. According to the study conducted by [11], the use of VSM for identification and selection of topics from Arabic documents, helped to get best results comparable to other methods. Automatic text summarization is an effective method for selecting the Arabic characters and reducing the noise from the documents [6].

Moreover, Hmeidi et al. [12] studied the influence of raw text, khoja root-based stemmer and light stemming of Arabic text documents based on standard classifiers, such as NB, SVM, KNN, J48 and Decision Table classifiers. The results exhibited that the SVM and NB classifiers with light stemming provides better classification accuracy than other classifiers.The same conclusion was drawn up by Al-Badarneh [13] and Ayedh et al. [14] by using various pre-processing methods. Additionally, Al-Molegi et al. [15] and Khreisat [16] have proposed an approach to classify Arabic text documents based on the combination of N-grams with some similarity measures, including Manhattan, Euclidean distances and Dice. The overall results illustrated that the combination of tri-gram with Dice measure obtained a better performance.

Al-Anzi et al. [17] presented a method based on LSI with clustering techniques for Arabic text classification by grouping similar unlabelled documents into a pre-defined number of topics. The results revealed that this method is able to label the documents without any training data. In another work, Al-Anzi et al. [18] offered a technique for Arabic text classification based on Latent Semantic Indexing and cosine similarity. The results showed that the LSI features outperform significantly the TF-IDF. Also, these results demonstrated that the KNN with cosine measureand SVM attained the best performance. Even though the most works in the literature review have already achieved a good performance, Arabic is a rich language that requires effective text classification algorithms, dealing with different aspects of the language, such as vocabulary, morphology, and syntax. [18], addressed some of the challenges of the Arabic language. Additionally, the authors in [19] used the conventional TF-IDF for Arabic text classification by using a number of different machine learning classifiers.

3. PROPOSED METHODOLOGY

Our proposed methodology aims to develop an automatic ATC system based on a kernel Naïve Bayes classifier. It consists of three steps as shown in Figure 1.

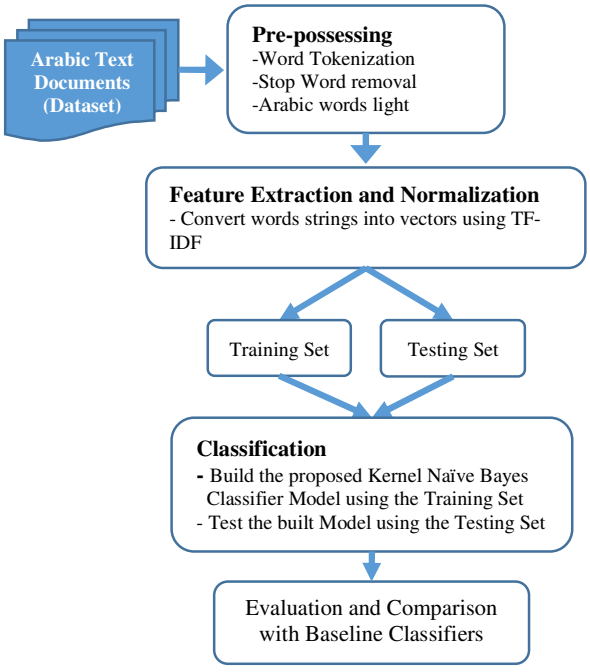

Figure 1. Proposed approach Methodology

The first step is a preprocessing step to extract the words (words tokenization), eliminate the stop words (stop words removal) from the collection of documents and remove common affix (prefix and suffix) from words (Arabic words light stemmer). The second step is a feature extraction and normalization step to convert the words strings into vectors and normalizing them. The third step is the classification step to classify those text documents into one of a pre-defined set of classes (topics) using the proposed kernel naïve Bayes classifier. In the following, we describe each step in more details:

3.1 Pre-processing

This step is responsible for Arabic words tokenization, stop words removal and Arabic words light stemmer. In tokenization, each sentence in documents is broken into tokens (words strings). Stop words removal is used to remove the useless words like "من" (from), "على" (on), etc. remove the stop word from documents increase the accuracy of the classification task.

3.2 Feature Extraction and Normalization

Feature extraction of words strings is based on the standard TF-IDF method that is one of the popular methods used in several text domains. TF-IDF is more efficient for selecting significant words that assigns high weights to the high frequency terms in different documents, but relatively rare in the whole corpus. The classical formula of TF-IDF is shown in the following Equation (1):

$$w_{x,y} = tf_{x,y} \log\left(\frac{N}{df_x}\right), \qquad (1)$$

Where $w_{x,y}$is the weight for word x in document y, $tf_{x,y}$ is the frequency of word x in document y, N is the number of documents in the collection, and df_x is the number of documents that contain the word x. Word frequencies for a document (instance) should be normalized by dividing them by document length.

3.3 Classification

In this step, we perform two tasks. The first task is usually building the model of machine learning by a selected dataset, called a training dataset. The second task is testing the built model using another unseen dataset, called a testing dataset. The proposed model used on our methodology is a KNB classifier. Actually, KNB is a Naïve Bayes with kernel density estimation (KDE). The following subsection presents an explanation of the proposed KNB.

3.3.1. Kernel Naïve Bayes (KNB) Classifier

Suppose X is a set of Arabic documents words features, $(x_i = x_1, x_2 .... x_N)$ and $C$is a set of Arabic topics (c_j) or classes. By Naive Bayes assumption, the probability ofa topic is c, given the features of words $x_1, x_2 .... x_n$ canbe calculated by the following Equation (2) and (3):

$$\begin{aligned} c &= \max_{c_j \in C} P(c_j \mid x_1, x_2 .... x_n) \\ &= \max_{c_j \in C} \frac{P(x_1, x_2 .... x_n \mid c_j) P(c_j)}{P(c_1, c_2 .... c_n)} \\ &= \max_{c_j \in C} P(x_1, x_2 .... x_n \mid c_j) P(c_j) \end{aligned} \qquad (2)$$

More generally:

$$P(x_1, x_2, x_3, ..., x_n \mid c_j) = \prod_i P(x_i \mid c_j) \quad (3)$$

The probability, $P(x_i \mid c_j)$, that the feature value of the word i is equal to x_i given the topic j (class j) is equal c_j, were estimated using KDE from a set of labeled training data (X, C). KDE is

$$P(x_i \mid c_j) = \frac{1}{Nh} \sum_{v=1}^{N} guKernel(x_i, x_{vi}), \quad guKernel(a,b) = \frac{1}{\sqrt{2\pi}} e^{\frac{(a-b)^2}{2h^2}} \quad (4)$$

a non-parametric way of estimating the probability density function population [20]. The probability $P(x_i \mid c_j)$, was estimated using Equation (4).

where *guKernel* is a Gaussian function kernel with variance 1 and mean zero, N is the number of the input data X belonging to class j which is equal c_j, $x_{vi}c$ is the feature value of the word in the i-th position of the v-th input $X = (x_{1i}, x_{2i} \ldots x_{Ni})$ in class j, and h is a bandwidth, or a smoothing parameter. To optimally estimate the conditional probabilities, h was optimized on the training dataset.

4. EXPERIMENT AND RESULTS

4.1 Dataset Collection

We created an Arabic topic mining corpus dataset that contains 1897 documents belonging to 3 different topics Economic (625 documents), culture (639 documents) and sport (633 documents). The corpus contains 2478 unique words. Table 1 shows the statistics of the createdcorpus dataset.Particularly, thedataset is collected from multiple online newspapers atdifferent time periods,from May to Jun of 2017 for our experiment of Arabic text classification. These different time periods of collecting the data make it more diversity to give a fair test of the classifier and more accurate evaluation of the work.

Table 1: Data Collection Statistics.

Name of Category	Number of Documents
Sport	448
Economic	405
Culture	563
Total	1897

4.2 Tool Description

The experiment was conducted using Waikato Environment for Knowledge Analysis (WEKA) [21].It is widely used for machine learning and data mining and originally developed at the University of Waikato in New Zealand.

4.3 Evaluation Measures and Comparison Results

The following five measures are computed to evaluate the performance of the proposed KNB classifier, using counts of true positives (TP; for the predicted topics which are correctly

classified as actual topics), false positives (FP; for the predicted topics which are incorrectly classified as actual topics), true negatives (TN; for the unpredicted topics which are correctly classified as actual topics) and false negatives (FN; for the unpredicted topics which are incorrectly classified as actual topics).

- Recall, or sensitivity, measures the proportion of the number of True Positives divided by the number of True Positives and False Negatives and is defined as TP/ (TP + FN).

- Precision measures the proportion of the number of True Positives divided by the number of True Positives and False Positives and is defined as TP/ (TP + FP).

- Accuracy (ACC) is the proportion of the number of True Positives and True Negatives divided by the number of True Positives, False Negatives, True Negatives and False Positives, and is defined as (TP + TN)/ (TP + FN + TN + FP).

- MCC indicates the degree of the correlation between the actual and predicted classes. MCC values range between 1, where all the predictions are correct, and −1 where none are correct, defined as ((TP × TN) −(FP × FN))/✓ (TP + FP) × (TP + FN) × (TN + FP) × (TN + FN).

- F-measure combines precision and recall into their harmonic mean, and is defined as 2 × (precision × recall)/ (precision + recall).

In our experiment, we split randomly the dataset into 70% as a training dataset and the remaining 30% are used as a testing dataset. Table 2 shows the number of instances in training and testing dataset.

Table 2: the number of instances in training and testing dataset.

No. of Instances in Training dataset	**No. of Instances in Testing dataset**	**Total**
1327documents	570 documents	1897 documents

During the evaluation phase, we computed the accuracy of training and testing, regarding to the proposed KNB classifier and NB classifier. The result shows that the KND achieved improvements of 13.1123% for the training accuracy and 9.4737% for the testing accuracy, compared to the traditional NB as seen in Table 3 and Figure 2.

Table 3: The results of training and testing accuracy for KNB and NB classifiers.

Accuracy / **Classifiers**	**Training Accuracy (%)**	**Testing Accuracy (%)**
Naïve Bayes (NB)	85.7573	81.7544
Proposed Kernel Naïve Bayes (KNB)	**98.8696**	**91.2281**

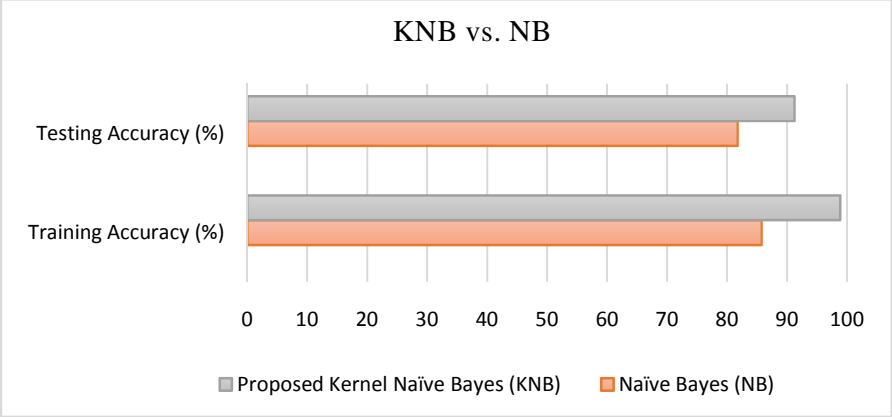

Figure 2. Results of KNB and NB classifiers.

Table 4 summarizes the results of KNB compared to the other baseline classifiers in terms of evaluation measures used in our experiment.

Table 4: The comparison results of the proposed KNB classifier and other baseline classifiers.

Classifier	**Accuracy (%)**	**Measure**	**Category**			
			Sport	**Economic**	**Culture**	**Average**
HMM	32.1053	Precision	0.000	0.000	0.321	0.103
		Recall	0.000	0.000	1.000	0.321
		F-Measure	0.000	0.000	0.486	0.156
		MCC	0.000	0.000	0.000	0.000
KNN	46.8421	Precision	0.704	0.976	0.378	0.468
		Recall	0.289	0.216	0.923	0.432
		F-Measure	0.410	0.353	0.537	0.309
		MCC	0.306	0.385	0.233	0.628
J48	73.8596	Precision	0.877	0.757	0.632	0.758
		Recall	0.685	0.737	0.798	0.739
		F-Measure	0.769	0.747	0.705	0.741
		MCC	0.679	0.623	0.550	0.619
SVM	85.7895	Precision	0.718	0.933	0.718	0.885
		Recall	0.945	0.874	0.945	0.858
		F-Measure	0.816	0.902	0.816	0.861
		MCC	0.727	0.857	0.727	0.802
NB	81.7544	Precision	0.952	0.744	0.787	0.830
		Recall	0.797	0.889	0.765	0.818
		F-Measure	0.867	0.811	0.776	0.819
		MCC	0.813	0.710	0.672	0.733
BN	84.2105	Precision	0.987	0.901	0.706	0.868
		Recall	0.751	0.863	0.918	0.842
		F-Measure	0.853	0.882	0.798	0.845
		MCC	0.806	0.825	0.698	0.777
Proposed KNB	**91.2281**	Precision	0.964	0.909	0.862	**0.913**
		Recall	0.954	0.895	0.885	**0.912**
		F-Measure	0.959	0.902	0.873	**0.913**
		MCC	0.938	0.853	0.812	**0.869**

As we see, the proposed KNB classifier achieves the highest accuracy (91.2281%), since its ability to solve the non-linearity problem of Arabic text classification. The highest and best results are highlighted as bold font in the Table 4. On the other hand, the HMM classifier achieved the lowest accuracy (32.1053%), as well as it has the lowest results for the other measures.The best MCC measure is obtained for the proposed KNB classifier with 0.869,whereas the worst is obtained for HMM with 0. The reason is that the most of instances are classified as correct for KNB compared to all baseline classifiers in our study. It is also clear that SVM classifier achieved a good result (85.7895%) compared to other classifiers. The result of SVM agree with the results of previous studies which found that it is a good classifier for text classification. In more details, we show that the Precision, Recall and F-Measure of our method are the highest with 91.3%, 91.2% and 91.3%, respectively, where as the Precision, Recall and F-Measure of the SVM are 88.5%, 85.8% and 86.1%, respectively. Generally, the proposed KNBclassifier attains the highest results in all measures of the study.

In addition to the evaluation of accuracy, the time taken to build the model for each classes is also studied and evaluated in our experiment. The comparison of this time is shown in Table 5.

Table 5: The time taken to build each classifier model in our experiment.

Classifier	Time taken to build model in seconds
HMM	**0.01**
KNN	0.32
J48	24.53
SVM	1.31
NB	0.82
BN	1.54
Proposed KNB	0.79

As shown in Table 5, the time taken to build the HMM model is the lowest with 0.01second, whereas the time taken to build the J48 model is the highest with 24.53seconds. However, the time taken to build our KNB modelhas0.79second compared to the time taken to build the SVM and NB which have 1.31seconds and 0.82second, respectively. In general, the time taken to build our model on training data is less than one second and almost near to the lowest ones with respect to the training speed.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a new approach for Arabic text classification using a Kernel Naïve Bayes (KNB) classifier. Some text pre-processing techniques, including word tokenization, stop word removal and Arabic words light stemmer are used. For Arabic words feature extraction, TF-IDF technique is also applied to convert them into vector space which are normalized for classification task. An effective classifier is proposed for classification in our approach. Experimental results on the collected dataset slows that our approach based on the proposed classifier achieved outstanding results in terms of accuracy and time against all baseline classifiers used in the previous studies.

The main conclusion is that Arabic text classification of electronic documents using the proposed KNB show better performance than other baseline classifiers in machine learning field. An open path of future research is to compare the performance of Arabic text classification classifiers with dimensionality reduction methods such as feature selection methods and topic models.

ACKNOWLEDGEMENTS

The authors would like to thank the reviewers for its valuable comments and time for reviewing the manuscript. Also, the authors would like to thank the faculty of computer and information science in Prince Sultan University (PSU) for supporting this work.

REFERENCES

[1] Sebastiani, F., (2002)"Machine learning in automated text categorization",ACM computingsurveys (CSUR), Vol. 34, No. 1, pp.1-47.

[2] Said, D., Wanas, N.M., Darwish, N.M. and Hegazy, N., (2009) "A study of text preprocessing tools for arabic text categorization", In:The second international conference on Arabic language, pp. 230-236.

[3] Abdulla, N.A., Ahmed, N.A., Shehab, M.A., Al-Ayyoub, M., Al-Kabi, M.N. and Al-rifai, S., (2014) "Towards improving the lexicon-based approach for arabic sentiment analysis", International Journal of Information Technology and Web Engineering (IJITWE), Vol. 9, No.3, pp.55-71.

[4] Beal V., (2011), "Search Engine",http://www.webopedia.com/TERM/S/search_engine.html Last visit on April, 2017.

[5] Al-Shargabi, B., Olayah, F. and Romimah, W.A., (2011) "An experimental study for the effect of stop words elimination for arabic text classification algorithms", International Journal of Information Technology and Web Engineering (IJITWE), Vol. 6, No. 2, pp.68-75.

[6] Al-Thwaib, E., (2014) "Text summarization as feature selection for arabic text classification", World of Computer Science and Information Technology Journal (WCSIT), Vol. 4, No. 7, pp.101-104.

[7] Dilrukshi, I., De Zoysa, K. and Caldera, A., (2013)"Twitter news classification using SVM", In: Computer Science & Education (ICCSE), 2013 8th International Conference on, IEEE, pp. 287-291.

[8] Al-Shalabi, R. and Obeidat, R., (2008) "Improving KNN Arabic text classification with n-grams based document indexing", In: Proceedings of the Sixth International Conference on Informatics and Systems, Cairo, Egypt, pp. 108-112.

[9] Srivastava, A.N. and Sahami, M. eds., (2009) Text mining: Classification, clustering, and applications, CRC Press.

[10] Park, H.H., Park, J. and Kwon, Y.B., (2015)"Topic clustering from selected area papers", Indian Journal of Science and Technology, Vol. 8, No. 26.

[11] Abainia, K., Ouamour, S. and Sayoud, H., (2015) "Neural Text Categorizer for topic identification of noisy Arabic Texts", In: Computer Systems and Applications (AICCSA), 2015 IEEE/ACS 12th International Conference of, IEEE, pp. 1-8.

[12] Hmeidi, I., Al-Ayyoub, M., Abdulla, N.A., Almodawar, A.A., Abooraig, R. and Mahyoub, N.A., (2015) "Automatic Arabic text categorization: A comprehensive comparative study", Journal of Information Science, Vol. 41, No. 1, pp.114-124.

[13] Al-Badarneh, A., Al-Shawakfa, E., Bani-Ismail, B., Al-Rababah, K. and Shatnawi, S., (2017) "The impact of indexing approaches on Arabic text classification", Journal of Information Science, Vol. 43, No. 2, pp.159-173.

[14] Ayedh, A., Tan, G., Alwesabi, K. and Rajeh, H., (2016), "The effect of preprocessing on arabic document categorization", Algorithms, Vol. 9, No. 2, p.27.

[15] Al-Molegi, A., IzzatAlsmadi, H.N. and Albashiri, H., (2015), "Automatic learning of arabic text categorization", Int. J. Digit. Contents Appl, Vol. 2, No. 1, pp.1-16.

[16] Khreisat, L., (2009), "A machine learning approach for Arabic text classification using N-gram frequency statistics", Journal of Informetrics, Vol. 3, No. 1, pp.72-77.

[17] Al-Anzi, F.S. and AbuZeina, D., (2016) "Big data categorization for arabic text using latent semantic indexing and clustering", In: International Conference on Engineering Technologies and Big Data Analytics (ETBDA 2016), pp. 1-4.

[18] Al-Anzi, F.S. and AbuZeina, D., (2017),"Toward an enhanced Arabic text classification using cosine similarity and Latent Semantic Indexing", Journal of King Saud University-Computer and Information Sciences, Vol. 29, No. 2, pp.189-195.

[19] Al-Anzi, F.S. and AbuZeina, D., (2015), "Stemming impact on Arabic text categorization performance a survey", In: Information & Communication Technology and Accessibility (ICTA), 2015 5th International Conference on, IEEE, pp. 1-7.

[20] Parzen, E., (1962) "On estimation of a probability density function and mode", in:The annals of mathematical statistics, Vol. 33, No. 3, pp.1065-1076.

[21] WEKA,"Data Mining Software in Java", http://www.cs.waikato.ac.nz/ml/weka,Last visit on May, 2017.

Authors

Mr. Raed Al-khurayji

Senior Software Consultant at Ministry of Communications and Information Technology of Saudi Arabia.

Dr. Ahmed Sameh Professor of Computer Science and Information Systems at Prince Sultan University