Efficient Topic Detection System for Online Arabic News

Mohammed M. Fouad
Faculty of Computing and Information Technology
King Abdulaziz University
Jeddah, Saudi Arabia

Marwa A. Atyah
Faculty of Media and Communication
King Abdulaziz University
Jeddah, Saudi Arabia

ABSTRACT
Nowadays, the news is updated very frequently, especially in the Middle East region where the Arabic language is the primary language of all its countries. The people in this region are interested in following up these updates through the available online news platforms. In order to automate the work in the news agencies, there is an urgent need for an automated system that is able to detect the topic of the news once it has arrived at the agency. In this paper, an efficient system is presented for classifying the online Arabic news into its proper topic. The proposed system uses various natural language processing techniques along with different classification methods. The experimental results show that utilizing the Information Gain, as a feature selection technique, with the Naïve Bayes algorithm, achieves the best accuracy in order to solve the topic detection problem for the online Arabic news.

General Terms
Data Mining, Natural Language Processing.

Keywords
Machine Learning, Text Mining, Arabic Online News, Topic Detection, Feature Selection.

1. INTRODUCTION
With the rapid growth of the internet and its services every day, many news outlets, such as newspapers or news agencies, published their updated news online on their official websites or other platforms using mobile applications. Most of the people use these online services as their source of the updated news because they could be reached anytime and anywhere.

The news articles on the website are mainly organized into many sections based on the different news topics as in the printed newspaper. These topics cover most of what the readers like and want to read such as Sports, Economy, Arts, Politics, Religion, etc. For the Middle East region, many news sources outlets, even the non-Arabic ones like BBC and CNN for example, published their Arabic news online because of the variety of the interesting topics in this region.

The news agencies are responsible for delivering the updated news to the news outlets. They also have their own platforms for publishing the news especially with the aid of mobile applications that are in the hands of most of the people nowadays. The news agencies and news websites are in a critical need to an automated system that is able to detect and classify the Arabic news into the appropriate topic based on its contents which can be considered a text classification task [1].

Due to the broad spread of the English language, most of the researchers in this area already published their own work with the support of the natural language processing toolkits that can handle the English language efficiently [2]. For the Arabic-based language contents, the researchers still face some problems in the computational linguistic parts of the research.

In this paper, an efficient topic detection system is proposed in order to classify the Arabic online news articles with very high accuracy. The proposed system utilizes different natural language processing techniques for analyzing Arabic contents with the support of feature selection techniques for better results.

This paper is organized as follows: Section 2 presents the related work in this research area especially in the Arabic language domain. Section 3 illustrates briefly the components of the proposed system. The experimental results are presented and discussed in Section 4. Finally, the conclusions are drawn in Section 5.

2. RELATED WORK
The problem of text document classification using different techniques has been addressed in many research studies. Most of these studies work on the English language because of the solid support of the tools and existence of different datasets. In this section, we present a number of the related research in the area of text document classification and highlight the different components in each approach.

Kanaan et al. in [3] presented three automatic text classification techniques using three different classification algorithms (Naïve Bayes, Rocchio, and K-NN). Their experiment compared the performance of the three algorithms on a manually collected dataset of 1,445 Arabic text documents that are classified into 9 categories. The results showed that Naïve Bayes algorithm outperforms other two algorithms with the best performance.

Vector space models are widely used in the Arabic text document classification. Ababneh et al. in [4] used K-NN algorithm with different variations of vector space models such as Cosine, Dice, and Jaccard coefficients. Their experiment was conducted on 5,121 Arabic documents collected from the Saudi Newspapers [5]. Using the average of precision, recall and F1 measures as a performance metric, their results showed that Cosine coefficients had the highest performance with all the different categories in the dataset.

Kompan and Bielikova proposed another vector representation in [6] for news articles but in the Slovak language. Their proposed model represented the feature vector differently by including words' collocations in the news article. Three different classifiers; such as Naïve Bays, K-NN, and Decision Tree, were used in the experiment to compare the performance of the proposed presentation. The experimental results showed that using words' collocations in the vector representation enhanced the pre-processing stage

but they were language dependent and should be changed greatly to work with other languages. In the same manner, Saad in [7] investigated the impact of the pre-processing stage on the text document classification, but in the Arabic language. Popular text classification algorithms, such as Naïve Bayes and Support Vector Machines, were used in the experiment. In addition, different techniques for Arabic morphological analysis and term weighting schemes were applied. Al-Saleem in [8] also utilized different steps in the preprocessing stage such as removing digits and punctuations, filtering all non-Arabic text and normalization of Hamza. These steps showed great improvement in the performance of the classification algorithms such as Support Vector Machines (SVM), and Naïve Bayes (NB).

Mesleh in [9] used chi-square as a feature selection method in order to reduce the generated feature vector for classifying Arabic documents. His experiment compared the performance of Support Vector Machines, K-NN, and Naïve Bayes algorithms when applied with the addition of chi-square method. The dataset used in this experiment contains about 1,400 Arabic news article. The result showed that F1 measure of Support Vector Machines was the highest when compared to other algorithms.

The Neural Networks were also applied on Arabic document classification. Harrag and El-Qawasmah in [10] collected 435 documents from the Hadith Encyclopedia that were classified into 14 categories. They applied the Neural Networks on this dataset to test the scalability problem of the features vector. The experimental results showed that Neural Networks achieved good results (about 88.3% accuracy) but the running time was very high because of the high dimensional problem of the text documents. Al-Tahrawi and Al-Khatib in [11] utilized the Polynomial Networks to be used for Arabic text documents. In their experiment, the classifiers were trained to classify the category in the one-versus-all method. They used the Alj-News dataset, that was presented in [12], for comparing the performance of Polynomial Networks (PN) with other algorithms such as Support Vector Machine, Naïve Bayes and J48. The experimental results showed that the performance of PN was not the best for all the categories in the dataset, but was very competitive. They also suggested some points for future work in order to improve the performance of PN.

The use of statistical classification methods, such as Maximum Entropy (ME) and Cumulative Thematic Probability (CTP), is also applied in the Arabic documents classifications. Swaaf et al. [13] and El-Halees [14] utilized ME method and showed its performance in the classification problem. Their experimental results showed that ME method can achieve good accuracy results (about 80% on average), but they did not compare ME method with the other commonly used algorithms such as Support Vector Machines, Naïve Bayes, K-NN and Decision Trees. On the other hand, Fodil et al. [15] used CTP function for text classification. Their proposed method builds a statistical model for each category in order to classify it efficiently.

Table 1 shows a summary of the discussed research work in this paper and summarizes for each study the used classification algorithms, datasets details and reference.

Table 1. Summary of Related Work for Arabic Text Classification

Reference	**Classification Algorithms**	**Dataset Characteristics**		
		Name	**# Documents**	**# Categories**
Kanaan et al. [3]	NB Rocchio K-NN	Newspapers websites	1,445	9
Ababneh et al. [4]	K-NN	Saudi Newspapers [5]	5,121	7
Kompan and Bielikova [6]	NB K-NN Decision Tree	SME.SK	1,387	20
Al-Saleem [8]	SVM NB	Saudi Newspapers [5]	5,121	7
Mesleh [9]	SVM NB	Newspapers websites	1,445	9
Harrag and El-Qawasmah [10]	Neural Networks	Hadith Encyclopedia	435	14
Al-Tahrawi and Al-Khatib [11]	Polynomial Networks	Aljazeera News	1,500	5
Swaaf et al. [13]	Maximum Entropy	Arabic NEWSWIRE	33,000	10, 34
El-Halees [14]	Maximum Entropy	Manual Collection	NA	6
Fodil et al. [15]	Cumulative Thematic Probability	News Book	175	5

3. THE PROPOSED SYSTEM

In this section, we will describe in brief details the proposed system for online Arabic news classification. As shown in Figure 1, the proposed system is composed of four stages. It starts with the preprocessing stage followed by feature extraction and representation stage, feature selection stage and

ends with learning the classification model using the proper algorithm.

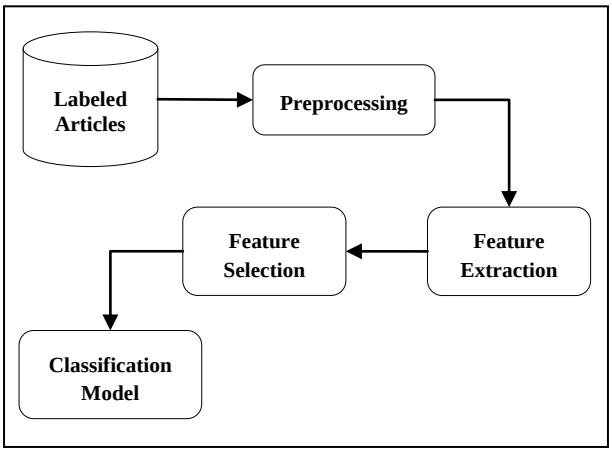

Fig 1: Overview of the Proposed System

In the proposed system, the Open Source Arabic Corpus (OSAC) provided by Saad and Ashour in [16] is used for building the classification model. OSAC corpus contains about 22,429 text documents that are manually categorized into 10 categories/topics as shown in Table 2. These documents were automatically collected from collected from different sources such as online news, blogs, free electronic books, etc.

Table 2. OSAC Dataset – Topic Distribution

Category/Topic	# Text Documents
Economic	3,102
History	3,233
Education & Family	3,608
Religious	3,171
Sport	2,419
Health	2,296
Law	944
Astronomy	557
Stories	726
Cooking Recipes	2,373
Total	**22,439**

3.1 Preprocessing

The proposed system started by applying the natural language processing techniques in order to process the text documents in OSAC dataset and create the appropriate feature vector for the classification task. In the preprocessing stage, we employed some steps such as normalization, tokenization, removing stop-words and light stemming for the Arabic language.

In the normalization step, some letters in the Arabic language are transformed into their base letter. For example, letters such as "أ", "إ" and "آ" are changed into "ا" only. In addition, all the non-Arabic letters are removed from the input text. This step is very important and reduces the number of generated words for vector representation stage. The tokenization step is responsible for splitting the input Arabic text into proper and meaningful text units, which are words. The result words are checked against stop-words, which are the words that were found commonly in the text and have no meaning with their own, such as "انا", "هم", "على" and "فوق" to be removed from the extracted words list. In the fourth step, which is light stemming, all the words are transformed into their root/stem word. For example, words such as "يلعبوا", "لعبة" and "لاعب" are changed into "لعب" which is the root word for them.

3.2 Feature Extraction and Representation

The generated words list, which is also called Bag of Words BoW, from the preprocessing stage are used to form the feature vector that represents each document in the input dataset.

There are several ways to represent the words/terms in the feature vector based on the weight of each word and its occurrence in the whole dataset. We have studied the effect of the different weighting schemas, shown in Table 3, to fill the bag-of-words extracted from the OSAC dataset. Consider the following annotations for describing the term weighting schemas. Let (a_{ik}) be the weight of term (i) in document (k), the total number of documents (N), Term frequency (f_{ik}) be the frequency of term (i) in document (k) and Document frequency (df_i) be the number of documents in which term (i) occurs.

Table 3. Term Weighting Schemas

Schema	Description
Binary	$a_{ik} = \begin{cases} 1, & f_{ik} > 0 \\ 0, & \text{otherwise} \end{cases}$
Term Frequency	$a_{ik} = f_{ik}$
Normalized Term Frequency	$a_{ik} = \frac{f_{ik}}{N}$
TF.IDF	$a_{ik} = f_{ik} * \log(\frac{N}{df_i})$

The basic schema is the binary schema in which the weight of the term in a certain document is calculated by its existence in this document or not. The most complex one is the TF.IDF schema that is calculated as a function of the term frequency and the number of documents that contains this term. The size of the generated feature vector is the same in each schema as the number of extracted words/terms still the same.

3.3 Feature Selection

As discussed earlier, each document is represented by a vector of numbers based on the extracted features. The dimension of this vector increased dramatically by the number of distinct terms in the input dataset collection. The curse of high dimensionality exists in most of the text processing systems including sentiment analysis ones. For this case, we used Information Gain (IG) as feature selection technique to reduce the dimension of the output feature vector. In the proposed system, the information gain weight is calculated for each feature and the features that have the higher weight than a predefined threshold are selected.

Consider the input Arabic text dataset with class attribute C that has divided into N classes $\{C_1, C_2, C_3, \ldots, C_N\}$. For any given feature x, the information gain (IG) is calculated by:

$$IG(x) = -\sum_{j=1}^{N} P(C_j)\log\left(P(C_j)\right) + P(x)\sum_{j=1}^{N} P(C_j|x)\log\left(P(C_j|x)\right) + P(\bar{x})\sum_{j=1}^{N} P(C_j|\bar{x})\log\left(P(C_j|\bar{x})\right) \quad \text{Eq. (1)}$$

Where, $P(C_j)$ is the fraction of documents labeled with class C_j, $P(x)$ is the fraction of documents in which feature x occurs and $P(C_j|x)$ is the fraction of documents with class C_j that has feature x.

3.4 Classification Model

The final stage is responsible for learning the classification model from the labeled feature vectors after the feature selection stage. In this stage, three of the commonly used algorithms in the text classification domain are implemented such as Support Vector Machines (SVM), Decision Trees (DT) and Naïve Bayes (NB). The OSAC dataset is used with 10-fold validation, in which the dataset is divided into 10 partitions with a stratified sampling of the classes. Each partition is used for testing and the other 9 partitions are used for training the classification algorithm.

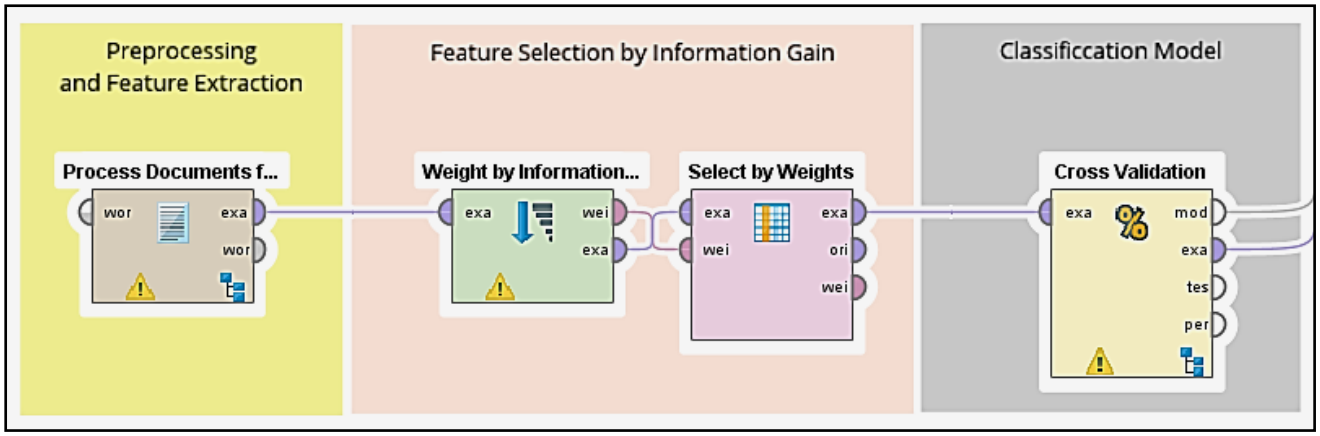

Fig 2: Implementation of the Proposed System using RapidMiner Studio®

4. RESULTS AND DISCUSSION

The stages of the proposed system were implemented using RapidMiner Studio® tool for data mining [17] on a machine with Intel® Core™ i7-4510U 64-bit processor (2.60 GHz), 8.00 GB memory and running Windows 8© operating system. Figure 2 shows the implementation of the proposed system in RapidMiner® tool with the support of Text Mining Extension.

As shown in Figure 2, both the preprocessing and features extraction and presentation stages are implemented in the "Process Documents from Files" operator. This operator is responsible for reading the Arabic text documents in each category, applies the preprocessing steps as illustrated earlier and outputs feature vectors with one of the term weighting schemas shown in Table 3.

The feature selection stage is performed using two operators. The first operator, Weight by Information Gain, calculates the value of the information gain for each term/attribute in the generated feature vectors. The second operator, Select by Weights, checks the calculated weights of the attributes and selects only the attributes with the higher weight than a predefined threshold value in the operator.

The final stage for building the classification model is implemented using the "Cross Validation" operator which is responsible for split the dataset into two sections for training and testing using 10-folds cross validation. This operator is also responsible for calculating the overall performance of the classification model when using SVM, DT or NB classifiers. The performance of the classifiers is determined using both the accuracy measure and the average F1-measure that is calculated using the average recall and precision from each individual class as follows:

$$Avg.F1 - Measure = 2 * (Avg.Precision * Avg.Recall)/(Avg.Precision + Avg.Recall)$$

4.1 Performance of the Classifiers

In the first experiment, we are interested in examining the performance of the individual classifiers when applied with the different term weighting schemas. First, the OSAC dataset is processed and represented with the term weighting schemas shown in Table 3. Second, for each schema, the classifiers are trained and tested with 10-folds cross validation without the feature selection stage. The accuracy, average recall, average precision and average F1-measure are reported as summarized in Table 4.

As shown in Table 4, the Naïve Bayes (NB) algorithm achieves the best performance and overcomes other two algorithms in all the used weighting schemas especially when implemented with TF.IDF schema. The accuracy of NB reaches about **94.19%** with F1-measure as **93.03%**. The results also show that Support Vector Machines (SVM) has a moderate performance. On the other hand, the Decision Tree (DT) algorithm has the worst performance with the different term weighting schemas, but its accuracy reaches about 83.03% when used with Binary schema.

Table 4. Comparison of Different Classifiers' Performance

Schema	Support Vector Machines		Naïve Bayes		Decision Trees	
	Accuracy	**Avg. F1**	**Accuracy**	**Avg. F1**	**Accuracy**	**Avg. F1**
Binary	84.22%	83.56%	88.17%	84.48%	83.03%	83.41%
Term Frequency	74.15%	72.91%	93.01%	91.96%	59.19%	46.58%
Norm. Term Frequency	80.89%	76.29%	85.73%	81.91%	76.63%	74.76%
TF.IDF	86.36%	84.08%	**94.19%**	**93.03%**	53.57%	50.14%

4.2 Using the Information Gain

The aim of this experiment is to show the effect of using the Information Gain (IG) technique for feature selection. In this experiment, the Naïve Bayes (NB) algorithm is used in building the classification model as it achieved the best performance as discussed earlier with the Binary term weighting schema.

Figure 3 shows the accuracy of the Naïve Bayes (NB) algorithm with the different weight thresholds for feature selection by the Information Gain technique.

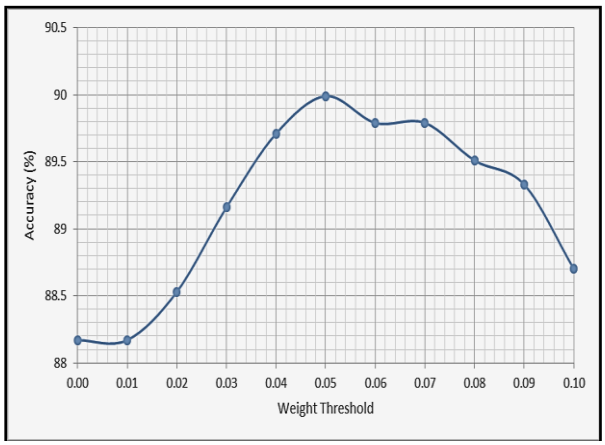

Fig 3: Accuracy of NB Algorithm with Different Weight Threshold for Information Gain

As shown in Figure 3, the Information Gain (IG) technique has a positive impact on the accuracy of the Naïve Bayes algorithm. The accuracy increased from 88.17% to reach 90.03% at 0.05 weight threshold value. Increasing the weight threshold above this value decreases the accuracy of the Naïve Bayes algorithm because the selection attributes were not sufficient enough to help the classifier to differentiate between the classes in the OSAC dataset.

5. CONCLUSIONS AND FUTURE WORK

In this paper, an efficient system for automatic topic detection of the online Arabic news was presented that could be used by the news agencies for classifying the input Arabic news efficiently. The proposed system utilized several natural language processing techniques along with the most commonly used algorithms for Arabic text documents classification. Feature selection by the Information Gain technique was also employed within the design of the proposed system.

The experimental results showed that Naïve Bayes (NB) algorithm achieved the best performance with about **94.19%** accuracy when applied with TF.IDF term weighting schema overcoming the Support Vector Machines and Decision Tree algorithms. The results also showed that using the Information Gain (IG) technique for the feature selection, not only reduced the number of features/attributes but also improved the performance of the base classifier.

As a future work, we are interested in utilizing the classifier ensemble method for Arabic text document classification [18]. In the classifier ensemble, the decisions of multiple classifiers are combined together to generate a single classifier decision. This method can benefit from the properties of the individual classifiers by the different methods for building such ensemble [19].

6. REFERENCES

[1] Lewis, D. 1991. Evaluating text categorization. In Proceedings of the Workshop on Speech and Natural Language (HLT '91; Vol. 91. pp. 312–318). Association for Computational Linguistics, Stroudsburg, PA.

[2] Aggarwal, C. and Zhai, C. (Eds.) 2012. Mining text data. New York: Springer USA.

[3] Kanaan, G., Al-Shalabi, R., Ghwanmeh, S. and Al-Ma'adeed, H. 2009. A comparison of text-classification techniques applied to Arabic text. Journal of the American Society for Information Science and Technology, 60(9), 1836–1844.

[4] Ababneh, J., Almomani, O., Hadi, Q., Kamel, N. El-Omari, T. and Al-Ibrahim, A. 2014. Vector Space Models to Classify Arabic Text. International Journal of Computer Trends and Technology (IJCTT), vol. 7, no. 4, 219-223.

[5] Al-Harbi, S., Almuhareb, A., Al-Thubaity, A., Korsheed, M. S. and Al-Rajeh, A. 2008. Automatic Arabic Text Classification. JADT'08: 9es Journées internationales d'Analyse statistique des Données Textuelles.

[6] Kompan, M. and Bieliková, M. 2011. News Article Classification Based on a Vector Representation Including Words' Collocations. In Advances in Intelligent and Soft Computing. Vol. 101, Berlin: Springer, 1-8.

[7] Saad, M. 2011. Arabic text classification. Saarbrücken, Germany: Lap Lambert Academic Publishing, VDM.

[8] Al-Saleem, S. 2011. Automated Arabic text categorization using SVM and NB. International Arab Journal of e-Technolgy, vol. 2, no. 2, 124–128.

[9] Mesleh, A.M. 2006. Chi square feature extraction based SVMs Arabic language text categorization system. Journal of Computer Science, vol. 3, no. 6, 430–435.

[10] Harrag F. and El-Qawasmeh, E. 2009. Neural Network for Arabic text classification. In Proceedings of the 2nd International Conference of Applications of Digital Information and Web Technologies, ICADIWT'09, pp.

778-783.

[11] Al-Tahrawi, M.M. and Al-Khatib, S.N. 2015. Arabic text classification using Polynomial Networks. Journal of King Saud University - Computer and Information Sciences, vol. 27, no. 4, 437-449.

[12] Khreisat, L. 2006. Arabic Text Classification Using N-Gram Frequency Statistics: A Comparative Study. In Proceedings of the 2006 International Conference on Data Mining (DMIN 2006), June 26–29, Las Vegas, Nevada, USA, pp. 78–82.

[13] Sawaf, H., Zaplo, J. and Ney, H. 2001. Statistical classification methods for Arabic news articles. In the Proceedings of the Arabic Natural Language Processing Workshop, ACL'2001, Toulouse, France, pp. 127–132.

[14] Fodil, L., Sayoud, H. and Ouamour, S. 2014. Theme classification of Arabic text: a statistical approach. In Terminology and Knowledge Engineering 2014, Berlin, Germany, pp. 77–86.

[15] El-Halees, A. 2006. Mining Arabic association rules for text classification. In Proceedings of the 1st International Conference on Mathematical Sciences, Palestine: Al-Azhar University of Gaza, pp. 157–167.

[16] Saad, M. K. and Ashour, W. 2010. OSAC: Open Source Arabic Corpus, Proceedings of the 6th International Conference on Electrical and Computer Systems (EECS'10), Lefke, North Cyprus, pp. 1-6.

[17] RapidMiner Studio® Tool for Data Mining Tasks and Implementations: https://rapidminer.com/

[18] Da Silva, N., Hruschka, E. and Hruschka Jr., E. 2014. Tweet sentiment analysis with classifier ensembles. Decision Support Systems, vol. 66, 170 – 179.

[19] Clark, S. and Wicentwoski, R.2013. SwatCS: Combining simple classifiers with estimated accuracy. In Proceedings of the seventh International Workshop on Semantic Evaluation, Atlanta, Georgia, USA, 425-429.