Automated Keyword Extraction using Support Vector Machine from Arabic News Documents

Conference Paper · April 2019

DOI: 10.1109/JEEIT.2019.8717420

2 authors:

Batool Alarmouty
Princess Sumaya University for Technology
3 PUBLICATIONS **2** CITATIONS
SEE PROFILE

Sara Tedmori
Princess Sumaya University for Technology
56 PUBLICATIONS **386** CITATIONS
SEE PROFILE

Automated Keyword Extraction using Support Vector Machine from Arabic News Documents

1st Batool Armouty
Computer Science Department
King Hussien School for Computing Sciences
Princess Sumaya University for Technology
Amman, Jordan
bat20178017@std.psut.edu.jo

2nd Sara Tedmori
Computer Science Department
King Hussien School for Computing Sciences
Princess Sumaya University for Technology
Amman, Jordan
s.tedmori@psut.edu.jo

***Abstract*—Keyword extraction is an indispensable step for many natural language processing and information retrieval applications such as; text summarization and search engine optimization. Keywords hold the most important information describing the content of a document. With the increasing volume and variety of unlabeled documents on the Internet, the need for automatic keyword extraction methods increases. Even though keyword extraction can be used in many applications, Arabic research in the field still lacking. In this paper, a supervised learning technique that uses statistical features and Support Vector Machine classifier was implemented and applied to extract the keywords from Arabic news documents. The proposed supervised learning approach achieved a precision of 0.77 and a recall of 0.58.**

***Index Terms*—Keyword Extraction, Arabic Language, Natural Language Processing, Support Vector Machine, Feature Extraction, Downsampling**

I. Introduction

The revolution of the Internet has made it very easy for individuals to search the web, create content and share it with others. The Internet has become the most popular source of information. Every day millions of people use search engines to search the web in a quest for information. Keywords in websites are not only useful for improving search rankings, but also very useful for website readers. Additionally, keywords are useful for various applications including text summarization, information retrieval, spam detection, indexing, clustering, and many others [1].

Keywords are the terms that best describe the documents and their content [2]. Keywords provide the reader with a general idea about the document content before reading it. Moreover, they are the link between the document and what the people are searching for. The more relevant and high quality the selected keywords are, the more they attract the right readers.

Although manual keyword extraction can be proficient, it is in-feasible because manual keyword extraction requires individuals with field knowledge, which is expensive, time-consuming, and tedious. The disadvantages of manual extraction bring the need to start automating the process. Automatic keyword extraction has attracted a considerable amount of attention primarily due to the usefulness it adds to a large number of applications[3].

Automatic keyword extraction is the process of identifying representative words that describe the content of the document with minimum human interaction [4]. Automatic keyword extraction can be performed using several techniques. These techniques can be generally divided into four classes [5]: (1) statistical analysis, which includes techniques like term frequency, (2) linguistic methods, like n-gram, (3) machine learning models, where a labeled set of documents is used for training the model to extract keywords like Nave Bayes [6], (4) hybrid approaches, which combine one or more techniques from the previous classes.

Automated keyword extraction from Arabic documents is facing many challenges. Arabic language is being used by a large percentage of the population around the world and is having an increasing number of publications and documents available on the internet, despite the limited amount of work done on the Arabic language. The Arabic language is also suffering from different dialects and lack of resources and support in the field.

In this paper, a supervised learning model that makes use of Support Vector Machine classifier, and statistical features extraction of Arabic news documents is introduced. The statistical features extracted from the documents are tf-idf and first occurrence. Each word in these documents along with its associated features is treated independently from the rest of the document. In order to predict whether the word is keyword or non-keyword efficiently, we balanced the number of non-keywords to the number of keywords after extracting the statistical features using downsampling.

The rest of this paper is structured as follows: Section II highlights the works related to the topic of this research. Section III presents the dataset used along with the preprocessing steps that have been carried out on it. Section IV discusses the feature extraction method applied to the dataset. Section V the employed Support Vector Machine classifier is discussed. Section VI lists the results of our model. Section VII contains limitations and future work. Finally, Section VIII concludes this research.

II. Related Work

Samawi et al [6] proposed an unsupervised learning method by applying a self-organizing map (SOM) neural network on Arabic documents. Their approach combines linguistic and statistical features to extract keywords from an individual document. This method was tested on two datasets, the best results were 42.84 precision, 46.79 recall, and the F1-measure was 44.72.

Awajan et al [7] combined statistical analysis and linguistics methods to extract keywords from documents. The documents were tokenized using morphological analysis and other natural language processing tools. Tokens with same stem or tokens with the same synonyms were grouped together. The author then computed the weight for each word using N-gram, after which, the documents became ready for the keyword extraction process. This experiment was tested on only three documents. The author attempted to extract a different number of keywords in each trial. The experimental results show an F1 measure of 0.4 for extracting 5 keywords, of 0.48 for extracting 10 keywords and an f-measure of 0.53 for extracting 15 keywords.

Amer and Foad [8] developed an unsupervised algorithm for keyword extraction called AKEA. The researchers used linguistic patterns based on POS tagging, statistical knowledge, and word-occurrence to extract the keywords. They also used Arabic Wikipedia as the third party to give the candidate keyword a confidence score if the candidate is indexed as a concept in Arabic Wikipedia. This algorithm was tested on four different datasets. The best result was an F1-measure of 0.298.

Sammak et al [9] used linguistic knowledge in addition to statistical features and syntactic rules using the POS tagging. The researchers used Linear Discriminant Analysis, but they used each word as it is instead of taking the stem. F1-measure was 0.38 for extracting 5 keywords and 0.49 for extracting 10 keywords.

Helmy et al [10] proposed a deep learning approach for the extraction of keyphrases from Arabic text, with a network architecture based on bidirectional Long Short-Term Memory (Bi-LSTM) Recurrent Neural Network. The authors applied their approach on a 6,000 document dataset that they build. The results attained were 0.216 and 0.419 for precision and recall, respectively.

Rammal et al [11] proposed an automated keyword extraction method using local grammar for developing an indexing system that extracts keywords automatically from Lebanese official journals using their titles, they focused on the first word of each title to determine which local grammar should be applied to suggest more potential keywords based on a set of features calculated for each node of the title. This research was done using 5,747 titles in which 76% of the manually extracted keywords were extracted automatically.

Najadat et al [12] proposed a novel unsupervised algorithm for Arabic keyphrases automatic extraction, they use attributes such as phrase position, phrase frequency, and phrase distribution. In their experiments, they tried many combinations of attributes and concluded that each feature alone produces poor results but when combining these features the results improved. The algorithm was tested on 200 documents obtained from Arabic Wikipedia and Food an Agricultural Organization of the United Nations, the best result obtained was 0.83 precision for extracting 10 keywords.

Abualigah et al [13] proposed a model for feature selection using Particle Swarm Optimization (PSO) to enhance text clustering using K-means algorithm. In order to improve the clustering, a subset of the features that contains the informative features is used by the k-means algorithm. For evaluation the method was tested on six different datasets and compared with other known algorithms, the proposed method achieve the best accuracy.

Abualigah et al [14] proposed a hybrid of Particle Swarm Optimization with genetic operators for the feature selection problem. This approach was tested on eight datasets, the result of the evaluation showed an enhancement on the K-means clustering than the previous feature selection method proposed by Abualigah et al [13] and other known approaches in terms of the text clustering accuracy, precision, recall, and F-measures.

III. Preprocessing

The preprocessing step is considered to be a critical step in natural language processing tasks since the way you transform the input document into features have an incredible influence on the final results. Documents consist of a group of sentences. The only thing needed to extract keywords is the words inside each document. Therefore, we need to get rid of any useless information within each document after splitting the sentences into its components: words, numbers, and punctuation.

Words can come in many different forms depending on various factors such as their Part of Speech and position in the sentence. Furthermore, words can have a different meaning when affixes are added to them, that is why we need to be careful while dealing with the input words. A word stem is defined as the base form of the word after removing affixes. Words with the same stem are considered to be in the same class, thus we need to group all words with the same stem together in order to have more accurate results.

The dataset used in this paper consists of 844 Arabic news documents crawled from aljazeera.net. Each document is associated with a keyword set that describes its content. On average, each document has 5 keywords.

The following preprocessing techniques were applied on the dataset; (1) tokenize the text into sentences, (2) all numbers were removed, (3) punctuation marks were removed, (4) each word was replaced with its stem, (5) the sentences were split based on space into tokens to create a bag-of-words for each document, (6) all the stop words were removed from the bag-of-words to prevent the bad influence they may have on the results, (7) finally the number of tokens in each document was calculated. Figure 1 shows the preprocessing steps.

Before creating the bag-of-words, all the words were stemmed. Afterward, words with similar stems within the

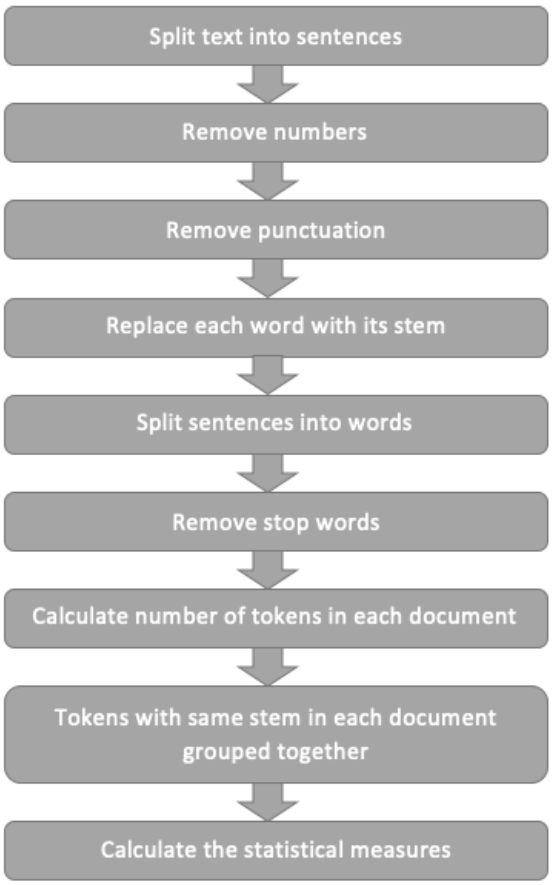

Fig. 1. Preprocessing steps applied on each document

bag-of-words were grouped together. The frequency of each stem was calculated and the position in which this stem first appeared in the document was stored. Then, tf-idf and first occurrence values of words - which will be explained in the next section - were calculated for each stem.

Regarding the keywords associated with the documents, if they consist of more than one word, they were first stemmed and then tokenized; otherwise, they were only stemmed. Figure 2 shows some of the keywords before and after preprocessing.

For all documents, each stem that matched a stem in the keywords associated with that document is labeled as "keyword" and given the value of "1". Otherwise, stems were labeled as "non-keyword" and given the value of "0". In the end, out of 159,842 Arabic words, 6,767 were labeled as keywords.

IV. Feature Extraction

After preprocessing was done, feature extraction was performed on the dataset. Feature extraction is the process of transforming the input data content into numeric features [15]. Machines cannot process text directly thus properties from the text were turned into numeric values. In this work, only the most related features to the keywords were extracted, where the relatedness of features to keywords was inferred from

Before stemming	After stemming
مياه	ماء
نهر الأردن	نهر ,أردن
أطفال	طفل

Fig. 2. keywords before and after stemming

previous works. Therefore, each word is represented by its numeric properties. In the experiments of our work, each word is represented by two features:

1) Tf-idf: which is the product of Term Frequency (TF) Inverse Document Frequency (IDF)[16], where TF represents the frequency of each word in the document. Since documents vary in length, the frequency of each word is divided by the length of the document:

$$TF(x) = \frac{Fx}{N} \tag{1}$$

Where, Fx: word x frequency.
N : number of words in the document. The IDF is calculated by taking the logarithm of the division of the whole number of documents in the corpus over the number of documents in which the word appears.

$$IDF(x) = \log \frac{D}{dx} \tag{2}$$

Where, D: number of documents in the corpus.
dx: number of documents word x appears in.

2) First Occurrence: this feature expresses how many words appeared before the current word being processed [17]. It is calculated by dividing the number of words before the current word by the total number of words in the document.

$$FC(x) = \frac{W}{N} \tag{3}$$

Where, W: number of words exist before words x.
N: number of words in the document.

Figures 3, 4 show the distribution of the variables.

V. Classification

In this paper, a Support Vector Machine that tries to find the best hyperplane which tries to separate the classes with the highest margin between the hyperplane and the data points was used, but the way the data is distributed, SVM will just classify in a dump way by considering all the words as non-keywords, as shown in figure 5, the number of non-keywords is way bigger than the number of the keywords, this kind of data is called imbalanced data. In such situation the SVM classifier will not try learning the differences in values between keywords and non-keywords, since considering all words as non-keywords will accomplish a very high accuracy which is what matters to the classifier, but in practice the classifier didn't learn how to identify the keywords from the given document and became useless for the keyword extraction task.

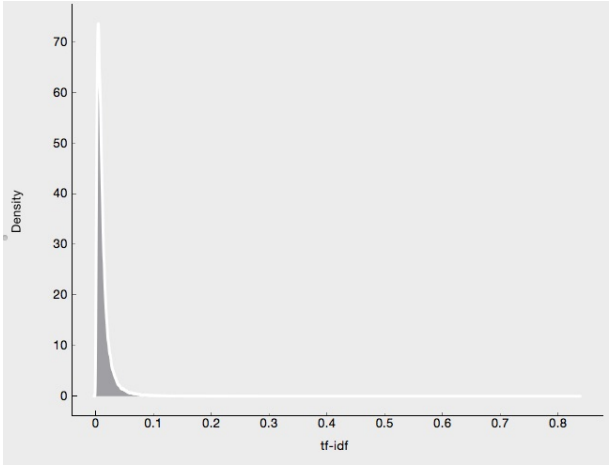

Fig. 3. tf-idf distribution.

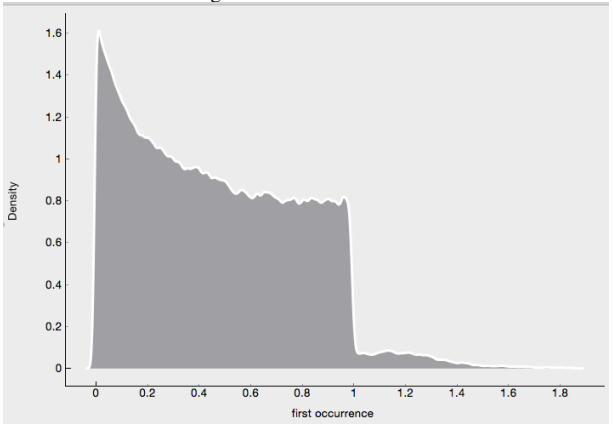

Fig. 4. First occurrence distribution.

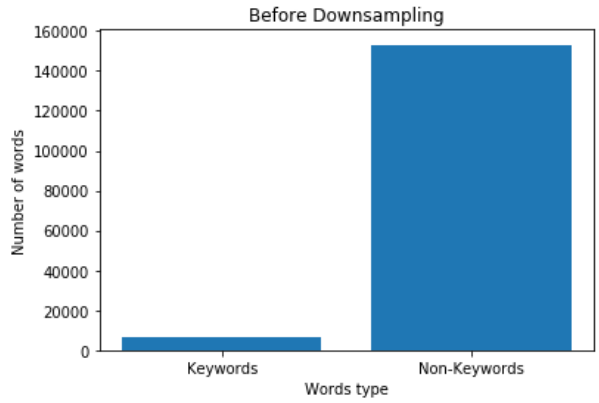

Fig. 5. Number of words in each class before applying down sampling.

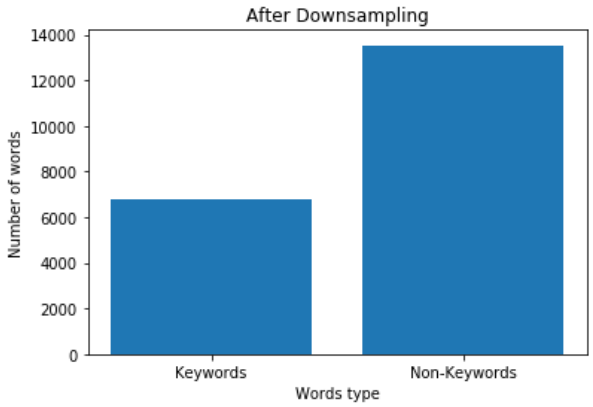

Fig. 6. Number of words in each class after applying down sampling.

To solve this problem, we tried to balance the data by making the number of non-keywords closer to the number of the keywords, to do that we used the downsampling method, which randomly takes samples from the class with the higher number of records, in our case is the non-keywords class. As shown in figure 6, the data is a lot more balanced than before which make it ready for the classification.

Before applying the classifier, the data values were to become between 0 and 1, this step is necessary for the data to become comparable. As for the kernel, the Radial Basis Function (RBF) was used, since our data is not linearly dependent, the SVM cant classify the data point without the kernel, using the RBF helps to find better hyperplane for separating the points classes.

VI. Results

For evaluating the success of our experiment, the precision (the percentage of selected keywords that are real keywords), recall (the percentage of the keywords that are selected), and F1 score (the tradeoff between precision and recall) were used to test our model performance. In order to be able to calculate each of these measures, the dataset was split randomly before classification into the train set and test set. The classifier trained on 80% of the dataset and tested on the rest, the ratio of the keywords to the non-keywords is the same in both training and testing sets.

After the classifier finished the training phase, we computed the confusion matrix using the test set to get the true positive, true negative, false positive and false negative values. These values are necessary to compute the evaluations measures mentioned before. In order to compare the results obtained from our classifier, we used both the Nave Bayes and Random Forest classifiers. As shown in figure 7, SVM classifier achieved better overall results than both the Nave Bayes and Random Forest.

VII. Limitations and future work

This research was conducted using only two statistical measures (tf-idf and first occurrence), in which may limit the results. Another weakness of this study is tokenizing the

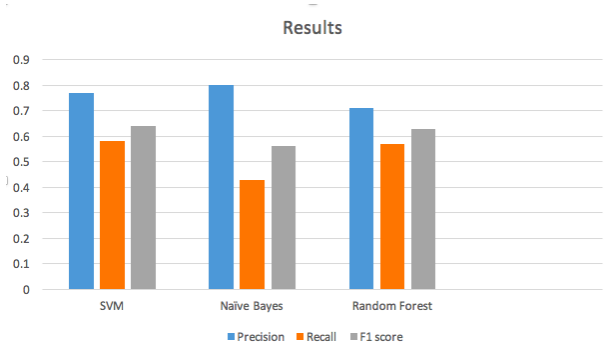

Fig. 7. The results obtained from the experiments as follow: Support Vector Machine: 0.77 precision, 0.58 recall and 0.64 F1 score, Nave Bayes: 0.8 precision, 0.43 recall and 0.56 F1 score, Random Forest: 0.71 precision, 0.57 recall and 0.63 F1 score. As explained the SVM achieved the best F1 score.

keywords that have more than one word into two words. In order to overcome these weaknesses, we can improve our approach by using other statistical measures. And apply the n-grams model to extract the keywords without the need to tokenize them.

VIII. Conclusion

In this work, statistical features (tf-idf and first occurrence) were extracted from an Arabic news dataset which consists of 844 documents crawled from Aljazeera.net. Each document was exposed to several necessary preprocessing methods required for extracting the features of each word in each document and make it easier for the classifier to process the data. After that each word in these documents was labeled as keyword or non-keyword based on the tags associated with each document, then the Support Vector Machine classifier was used for the classification of the keywords and resulted in an f-measure of 0.64, which is considered a good result in comparison with other Arabic automated keyword extraction methods, as its use a supervised machine learning technique to predict either the word is a keyword or non-keyword independently of the rest of the words in the document instead of trying to extract the keywords by generating keyword candidates and choose the most suitable words. Finally, by using the downsampling method to balance the data so the SVM classifier becomes more capable of distinguishing the keywords from non-keywords in a given input, which gave more strength to the final results.

References

[1] Cohen J., Highlights: language- and domain-independent automatic indexing terms for abstracting, Journal of the American Society for Information Science, 1995, pp. 162-174.

[2] Zhang K., Xu H. et al, "Keyword Extraction Using Support Vector Machine", Springer, 2006, pp. 85-96.

[3] Giarlo M., A Comparative Analysis of Keyword Extraction Techniques, CiteSeer, 2005, pp.1-14.

[4] Awajan A., Keyword Extraction from Arabic Documents using Term Equivalence Classes, ACM Trans, 2015, Vol. 14, No. 2, Article 7, pp. 7:1-7:18.

[5]Bharti S., Babu K. et al, Automatic Keyword Extraction for Text Summarization: A Survey, National Institute of Technology, 2017, pp. 1-12.

[6] Omoush E.,Samawi V., Arabic Keyword Extraction using SOM Neural Network, International Journal of Advanced Studies In Computer Science and Engineering, Volume 5, Issue 11, 2016, pp. 7-12.

[7] Awajan A., Suleiman D., Bag-of-Concept Based Keyword Extraction from Arabic Documents, International Conference on Information Technology (ICIT), 2017, pp. 863-869.

[8] Amer E., Foad K., AKEA : An Arabic Keyphrase Exraction Algorithm, Springer, 2017, pp.137-146.

[9] El-shishtawy T.A., Al-sammak A.K., Arabic Keyphrase Extraction using Linguistic Knowledge and Machine Learning techniques, arxlv, 2012.

[10] Helmy M., Vigneshram R. et al, Applying Deep Learning for Arabic Keyphrase Extraction, ScienceDirect, 2018, pp. 1-8.

[11] Rammal M., Bahsoun N. et al, Keyword extraction from Arabic legal texts, Interactive Technology and Smart Education, 2015, Vol. 12 Issue: 1, pp.62-71.

[12] Najadat H., Al-Kabi M. et al, Automatic Keyphrase Extractor from Arabic Documents, International Journal of Advanced Computer Science and Applications, 2016, Vol. 7, No. 2, pp. 192-199.

[13] L.M. Abualigah, et al., A new feature selection method to improve the document clustering using particle swarm optimization algorithm,2017, J. Comput. Sci. ,pp. 1-11, http://dx.doi.org/10.1016/j.jocs.2017.07.018.

[14] Abualigah L., Khader A., Unsupervised text feature selection technique based on hybrid particle swarm optimization algorithm with genetic operators for the text clustering, 2017, Springer, pp. 4773 4795.

[15] Saxena D., Saritha S. et al, Survey Paper on Feature Extraction Methods in Text Categorization, International Journal of Computer Applications, 2017, Vol. 166, pp. 11-17.

[16] Frank E., Witten I. et al, KEA: A Practical Automatic Keyphrase Extraction, ACM, 1999, pp.1-23.

[17] Zhang K., Xu H., et al, Keyword Extraction using Support Vector Machine, Springer, 2006, pp. 85-96.