International Journal of Engineering & Technology, 7 (3.28) (2018) 205-208

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

A Proposed Method for Reducing the Dimension of Arabic Documents

Aledinat Lowai Saleh, **Syed Abdullah Fadzli***

Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin, 21300 Kuala Terengganu, Terengganu, Malaysia
**Corresponding author E-mail: fadzlihasan@unisza.edu.my*

Abstract

Dimensionality reduction is an essential data preprocessing technique for large-scale and streaming data classification tasks. It can be used to improve both the efficiency and the effectiveness of documents. Traditional dimensionality reduction approaches fall into two categories: Feature Extraction and Feature Selection. Techniques in the feature extraction category are typically more effective than those in feature selection category. The representation of Arabic texts and the possibility of reducing their size result in reducing dimension among them, thus facilitating the processes and procedures that occur on them such as measurement of similarities, text classification, etc. Though several researchers have innovated many methods to solve this problem, in this paper, we introduce an effective method to represent Arabic texts with the lowest size. This method is based on the structure or form of words in Arabic, in terms of removing all prefixes and suffixes from words in the texts as well as removing the redundant and meaningless words. This methodological procedure could help in increasing the size of texts. Remove these prefixes and suffixes from words in the text aims at reducing dimension. The experimental results presented evidence that the proposed method substantially reduces the size of text representation by 42%, taking into account the origin of texts and words that reduce dimension.

Keywords: *Arabic Text Processing; Stop-word List; Text dimensionality reduction.*

1. Introduction

Knowledge and data have grown rapidly; therefore, it is challenging to extract correct information from texts. Natural language process aims at extracting information by classifying and organizing text files. Hence, the concentration must be on the possibility of reducing the size of a set of texts that can be accessed through Natural Language Processing (NLP). Unlike high-dimensional texts, NLP functions ideally with low-dimensional ones. Accordingly, the dimension-reducing methods of texts without losing basic data have a considerable impact on the time-and-space request uses, thus improving the quality of several science mission results. Natural language processing (NLP) is a branch of artificial intelligence science with some significant implications on how computers and humans interact. Having developed over the course of time, the human language has become a nuanced form of communication with rich information that transcends the boundaries of words. Hence, NLP with all its processes will be a vital tool in bridging the gap between human communication and the stored digital information. NLP's applications manipulate text documents as a set of separate words. Arabic texts are represented in two ways: Vector Representation (VR), and Graphical Representation (GR). The VR depends on Vector Space Model (VSM), whereby the component represents the diverse features of the text, mainly its terms [1]. The VSM represents a diversity of documents (datasets) that is used to measure the similarity and correspondence of natural language texts with a set of documents (n) containing terms (t) that are not repeated or unique. Each document Di is represented as Di =< di1, di2, dit> of dimension (m), that represents a vector in the vector space. The component (dij) represents the terms frequency or the weight of the (jth) term in the document Di. Therefore, all sets of documents can be represented as a term by document matrix of column vectors D. The (t) rows represent terms and the (n) columns represent documents. In the vector Di, the word order separately leads to the loss of the sentence or phrase meaning although the vectors are used to measure the similarity between the texts [2]. Reducing VSM dimensionality is one of the important areas of research in the information retrieval (IR) and NLP research communities. There are two different approaches for dimensionality reduction: language-independent approach and language-dependent one. Among the language-independent reduction strategies, the singular value decomposition and independent element analysis are the most common. These reduce the dimensionality of the vector space by providing a reduced rank approximation within the column and row space of the document matrix [3]. Dimension-reducing techniques are needed in order to reduce the computational time needed by the feature weight and classification stages. This would save the necessary storage. Consequently, proposing new techniques for enhancing the performance of feature filtering is inevitable. These techniques should not add complexity to the filtering approach in order to keep it simple and fast. In order to investigate the performance of these new techniques, a large comparative study has to be conducted using different benchmark datasets of different natures in terms of the vocabulary size and category distribution.

2. Related Studies

In [14] proposed different representations of classifier K-nearest neighbors (KNN) for Arabic texts that concentrated on the development of texts and classifiers rather than proposing a new or improved FS. The authors reported that the value of Macro-F1 reached 0.93. In [13] investigated the classification techniques with a large and diverse dataset. These techniques include a wide range of classification algorithms, feature selection methods, and representation schemes. Regarding feature selection, the best average result was achieved using the GSS method with TF as the base for calculations. Concerning feature weight functions, it was concluded that the length term collection (LTC) was the best performer. According to Boolean and Term Frequency Collection (TFC), the experiments showed that the SVM classifier outperformed other algorithms. In another study by [15], a binary FA for Feature Selection was developed. To test the efficiency of this method, Ghany used several benchmark datasets and compared them to two well-known bio-inspired methods (GA and PSO). The results showed that the proposed FA outperformed GA and PSO in improving the classification performance and reducing the feature set. Ghany's findings reported a classification error between 0.024 and 0.297. In [11] proposed various criteria for text mining. These criteria may be used to evaluate the effectiveness of the used text mining techniques. This makes the user choose one technique among the several available text-mining techniques. In [8] developed a new Arabic light stemmer called P-stemmer, that is a modified version of Larkey's light stemmers. Kanan showed that his stemming approach significantly enhanced the results for Arabic document categorization when using Naive Bayes (NB), SVM, and random forest classifiers. The experiments he used showed that SVM performed better than the other two classifiers. In [9] developed a hybrid FA based Feature Selection by combining it with Simulated Annealing to improve the obtained results. To validate this proposed technique, 11 regression and 29 classification datasets were used and compared with different existing methods, thus resulting in satisfactory findings. In [10] proposed a new Feature Selection method based on Ant Colony Optimization (ACO) for Sentiment Analysis. A k-NN classifier was employed to evaluate the performance of this proposed technique using customer review datasets. The results were compared with Information Gain (IG), Genetic Algorithm (GA) and Rough Set Attribute reduction (RSAR). A maximum precision of 0.892 was reported which was the best result. A further study by [12] significantly reduced the size of text representation by about 27 % compared with the stem-based VSM, and by about 50 % compared with the traditional bag-of-words model.

3. Methodology

In this paper, a new method is used to achieve a new representation of Arabic documents in a smaller size, thus ensuring the reduction of the dimension. This new method aims at reducing the dimension between texts by offering some rules. These rules have been developed to identify and remove repeated words, stop words, affixes including prefixes and suffixes in Arabic words, namely Tokenization, stop words Removal, Stemming.

3.1. Tokenization

Tokenizing means splitting your text into minimal meaningful units that is a mandatory step prior to other processing steps. In other words, tokenization is the task of breaking the text into pieces, called tokens, and at the same time, certain characters such as punctuation, numbers, non-Arabic character, and repeated words are neglected. The text in Arabic is often composed of a set of words that are repeated frequently. This rule was developed to identify these words and to adopt only one word. The following example represents tokenization.

Input:

1: مجلس الأمن الدولي يصوت على مشروع القرار -55- بفرض عقوبات تجارية.

Output:

مجلس الأمن يصوت على مشروع القرار يفرض عقوبات تجارية

3.2. Stop words removal

Stop words are common words that appear in the text carrying little meaning; they serve only syntactically rather than semantically. These stop words have two different impacts on the information retrieval process. They can affect the retrieval effectiveness because they have high frequency occurrences and diminish the impact [16]. Therefore, the proposed method in this paper delete these words from the texts, including the letters in Arabic language, pronouns and Linking letters. Consequently, this paper used -97- stop words removal similar to the sample shown in Table 1.

Table 1: List sample of stop words

عليها	عليه	هم	عن	لكن	هو	بن	عدد	اي	ب
هي	اخرى	او	حتى	غدا	إذا	بعض	ضد	ماذا	اف

3.3. Stemming

To obtain the root word in Arabic, affixes including prefixes and suffixes must be removed. Prefixes and suffixes are letters attached to the words either at the beginning (prefixes) or at the end (suffixes). Using these affixes do not affect the meaning of the root words. In this paper, prefixes are classified into three classifications depending on the number of characters as follows: (i) three-letters prefix (بال , وال), (ii) two-letter prefix (ال, لل) and (iii) one letter prefix (ت , ب , ا). The same classification is applied to suffixes; (i) three-letter suffixes (يان , تان , وهم), (ii) two-letter suffixes (ات , ين , ون), and (iii) one-letter suffix (ة , ه). The suffixes and prefixes used in this paper are shown in Table 2.

Table 2: List of prefixes and suffixes

3- letter prefix	بال	كال	وال	ولل							
2- letter prefix	ال	لل	سي	ست							
1- letter prefix	ا	ن	ي	ت	س	ل	و	ب			
3- letter suffix	تمل	همل	تان	يان	تين	كمل	كنا	وهم			
2- letter suffix	ين	ان	ون	وا	تم	هم	كم	ات	ها	نا	هن
1- letter suffix	ه	ة									

This paper proposes a method to remove the Arabic prefixes and suffixes by comparing the words with the prefixes and suffixes list. There are six steps in removing all prefixes and suffixes.

Step 1 (Remove 3 letters prefixes): Each token is compared with the 3 letter prefixes list. If the prefix is matched, it will be removed if the length of the word is greater or equal to six characters as in Table 3.

Table 3: Remove 3 letters prefixes

Token	كالمعلمين	والمدراس	كالملاعب	بالتدريس
Remove prefix	معلمين	مدراس	ملاعب	تدريس

Step 2 (Remove 2 letters prefixes): Each token is compared with the 2 letter prefixes list. If the prefix is matched, it will be removed if the length of the word is greater or equal to five characters as in Table 4.

Table 4: Remove 2 letters prefixes

Token	المعلمون	للمدارس	سيلعبون	ستكتبون
Remove prefix	معلمون	مدراس	لعبون	كتبون

Step 3 (Remove 1 letter prefix): Each token is compared with the 1 letter prefix list. If the prefix is matched, it will be removed if the length of the word is greater or equal to four characters as shown in Table 5.

Table 5: Remove 1 letters prefixes

Token	اسلام	نرجع	تكتب	يقول
Remove prefix	سلام	رجع	كتب	قول

Step 4 (Remove 3 letters suffixes): Each token is compared with the 3 letter suffixes list. If the suffix is matched, it will be removed if the length of the word is greater or equal to six characters, as in the following example.

Table 6: Remove 3 letters suffixes

Token	تستقبلوهم	نستقدمهما	استحقاقكما	يدفعهما
Remove prefix	تستقبل	نستقدم	استحقاق	يدفع

Stop 5 (Remove 2 letters suffixes): Each token is compared with the 2 letter suffixes list. If the suffix is matched, it will be removed if the length of the word is greater or equal to five characters, as Table 6 illustrates.

Table 6: Remove 2 letters suffixes

Token	يقولون	يدرسهن	يستلقيان	مدرسات
Remove prefix	يقول	يدرس	يستلقي	مدرس

Step 6 (Remove 1 letter suffix): Each token is compared with the 1 letter suffix list. If the suffix is matched, it will be removed if the length of the word is greater or equal to four characters. Table 7 introduces an example of step 6.

Table 7: Remove 1 letters suffixes

Token	مدرسه	خياطة	ايرانيه	ماليزية
Remove prefix	مدرس	خياط	ايراني	ماليزي

4. Results and discussion

This paper depends on a dataset that included 1737 Arabic documents [18] from the BBC News, divided into multiple topics, as shown in Table 8.

Table 8: Dataset of Arabic documents

Category of News	No. of Documents
Middle East News	868
Public World News	122
Economy and business	296
IT News	232
Sports news	219
Total	1737

After conducting the cutting process and removing excess symbols, the processed text represented less than the size of the original text. The examples below show some documents represented in Table 9 below, including the count words of each document before and after tokenizing.

Table 9: Tokenization example

DOC	Original Text	After Tokenization
D703	4390	2305
D92	3145	1606
D37	3130	1605
D720	1932	965
D9	1766	967
D704	1749	960

After deleting the suggested stop words [17], the text has a lower size. Some examples of some documents are represented in Table 10 below which shows the size of the words in each document before and after processing.

Table 10: Stop words removal example

DOC	Original Text	After Tokenization	Stop Words Removal
D703	4390	2305	2199
D92	3145	1606	1517
D37	3130	1605	1517
D720	1932	965	893
D9	1766	967	915
D704	1749	960	890

To maximize the efficiency of the Arabic words stemming, this paper proposes 3 arrangements of prefixes and suffixes removal as follows:

Arrangement A: Step 1 → Step 2 → Step 3 → Step 4 → Step 5 → Step 6.
Arrangement B: Step 6 → Step 5 → Step 4 → Step 1 → Step 2 → Step 3.
Arrangement C: Step 1 →Step 6 → Step 2 → Step 5 → Step 3 → Step 6.

After applying all the arrangements to the whole dataset, each arrangement shows different results as in Table 11.

Table 11: Stemming example

DOC	Original Text	After Tokenization	Stop Words Removal	Stemming		
				Arr A	Arr B	Arr C
D703	4390	2305	2199	1711	1823	1903
D92	3145	1606	1517	1166	1212	1400
D37	3130	1605	1517	1167	1200	1320
D720	1932	965	893	689	700	724
D9	1766	967	915	737	800	819
D704	1749	960	890	724	750	780

After applying all the steps to the whole dataset, different results were obtained as in Table 12.

Table 12: The whole document dataset

Total No. Doc	Total No. Words	After Tokenize	Stop Words Removal	Stemming		
				Arr A	Arr B	Arr C
1737	828116	617507	510344	480241	498201	501488

The proposed algorithm that has been tested focuses on the proposed dataset. The results compared the original texts without any processing with the processed texts. Each text is represented as a vector containing only unique words. Most studies in Arabic focus on reducing the dimension between a dataset and texts on the root extraction of words. Results after implementing the algorithm for the entire documents are as follows after reducing the dimension to 42% for each data set or dataset as illustrated in Table 13 and Figure 2.

Table 13: The number of words after processing (sample data)

Doc	Original Text	After Tokenization	Stop Words Removal	Stemming		
				Arr A	Arr B	Arr C
D703	4390	2305	2199	1711	1823	1903
D92	3145	1606	1517	1166	1212	1400
D37	3130	1605	1517	1167	1200	1320
D720	1932	965	893	689	700	724
D9	1766	967	915	737	800	819
D704	1749	960	890	724	750	780
D729	1697	859	805	639	660	710
D256	1650	851	773	632	671	694
D449	1476	895	832	713	768	780
D1526	1434	875	816	680	690	710

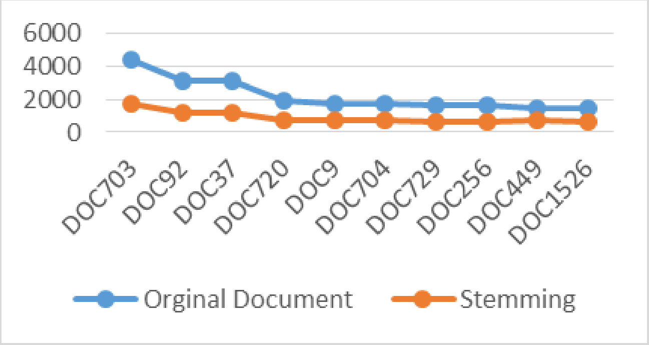

Fig. 2: Diagram appear method to reduce dimension (sample data)

5. Conclusion

This paper has presented an alternative technique for processing high-dimensional Arabic texts, which are often presented in reduced size. The formal similarity analysis is used to collect similar words to lead or generate one formal form. Experiments on different data sets (a set of text files) have shown that using similarities in the form reduces the size of datasets. Significantly, the result appears to reduce the dimension to about 42%; the reduction of the dimension in Arabic documents depends on the nature of the text and its class. Future work may focus on categorizing and summarizing datasets for obtaining better results.

References

[1] Salton, G., Wong, A., & Yang, C. S. (1975). A vector space model for automatic indexing. Communications of the ACM, 18(11), 613-620.

[2] Turney, P. D., & Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. Journal of Artificial Intelligence Research, 37, 141-188.

[3] Baker, K. (2013). Singular value decomposition tutorial. Note for NLP Seminar, pp. 1-24.

[4] Moh'd A Mesleh, A. (2007). Chi square feature extraction based SVMs Arabic language text categorization system. Journal of Computer Science, 3(6), 430-435.

[5] Al-Harbi, S., Almuhareb, A., Al-Thubaity, A., Khorsheed, M. S., & Al-Rajeh, A. (2008). Automatic Arabic text classification. Proceedings of the 9th International Conference on the Statistical Analysis of Textual Data, pp. 77-83.

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

[7] Saad, M. K., & Ashour, W. (2010). Arabic text classification using decision trees. Proceedings of the 12th International Workshop on Computer Science and Information Technologies, pp. 75-79.

[8] Kanan, T., & Fox, E. A. (2016). Automated Arabic text classification with P-S temmer, machine learning, and a tailored news article taxonomy. Journal of the Association for Information Science and Technology, 67(11), 2667-2683.

[9] Zhang, L., Mistry, K., Lim, C. P., & Neoh, S. C. (2018). Feature selection using firefly optimization for classification and regression models. Decision Support Systems, 106, 64-85.

[10] Ahmad, S. R., Yusop, N. M. M., Bakar, A. A., & Yaakub, M. R. (2017). Statistical analysis for validating ACO-KNN algorithm as feature selection in sentiment analysis. AIP Conference Proceedings, 1891(1), 1-7.

[11] Hashimi, H., Hafez, A., & Mathkour, H. (2015). Selection criteria for text mining approaches. Computers in Human Behavior, 51, 729-733.

[12] Awajan, A. (2016). Semantic similarity based approach for reducing Arabic texts dimensionality. International Journal of Speech Technology, 19(2), 191-201.

[13] Khorsheed, M. S., & Al-Thubaity, A. O. (2013). Comparative evaluation of text classification techniques using a large diverse Arabic dataset. Language Resources and Evaluation, 47(2), 513-538.

[14] Alhutaish, R., & Omar, N. (2015). Arabic text classification using k-nearest neighbour algorithm. International Arab Journal of Information Technology, 12, 190-195.

[15] Emary, E., Zawbaa, H. M., Ghany, K. K. A., Hassanien, A. E., & Parv, B. (2015). Firefly optimization algorithm for feature selection. Proceedings of the ACM 7th Balkan Conference on Informatics Conference, Article No. 26.

[16] El-Khair, I. A. (2006). Effects of stop words elimination for Arabic information retrieval: A comparative study. International Journal of Computing and Information Sciences, 4(3), 119-133.

[17] Rank NL. (2018). Arabic stopwords. https://www.ranks.nl/stopwords/arabic.

[18] Saad, M. K., & Ashour, W. (2010). Arabic morphological tools for text mining. Proceedings of the 6th International Conference on Electrical and Computer Systems, pp. 1-6.