Improving TF-IDF with Singular Value Decomposition (SVD) for Feature Extraction on Twitter

Ammar Ismael Kadhim[1], Yu-N Cheah[2], Inaam Abbas Hieder[3], Rawaa Ahmed Ali[4]

[1,3&4]Department of Computer Science,College of Medicine ,Universitiy of Baghdad,
[2]School of Computer Sciences,Universiti Sains Malaysia,Penang, Malaysia
[1]ammarusm70@gmail.com, [2]inaam_abbas@yahoo.com,[3] rawaa.ahmed@gmail.com,
[4]yncheah@cs.usm.my

doi: 10.23918/iec2017.16

ABSTRACT

Feature extraction is provided a lot of significance in social networks such as Twitter, due to playing a vital role in public opinion analysis. Several algorithms are suggested for solving them. Feature extractions are generally defined as to the process of extracting interesting features, non-trivial and knowledge from unstructured text documents. Feature extractions are interdisciplinary field which depends on information retrieval, machine learning, parameter statistics and computational linguistics. This study implements two methods term frequency-inverse document frequency (TF-IDF) and logarithm (TF-IDF) with singular value decomposition (SVD) dimensionality reduction techniques. The paper presents a new method that displays an effective preprocessing and dimensionality reduction techniques which help the feature extraction by using logarithm TF-IDF method. Finally, the experimental results show that logarithm TF-IDF method enhances the performance of English text document classification. Simulation results show the superiority of the proposed algorithm. In general, TF-IDF with logarithm outperforms traditional TF-IDF with respect to the evaluation metrics.

Keywords: Feature Extraction, Retrieval Information, Classification, Singular Value Decomposition, Dimension Reduction, TF-IDF.

1. INTRODUCTION

All before going to implement any operation on the feature extraction, the texts have to be processed. Due to the text often involves some special formats or non-informative features like date formats, number formats, and the most widely words that non-informative to help feature extraction like prepositions, articles, and pro-nouns can be removed. Dimension reduction is one of important process in text classification and enhances the performance of classification techniques via reducing dimensions so that text classification algorithms process text documents with a reduced number of features. Singular value decomposition (SVD) is a way used to reduce the dimension of a vector that is used in linear transformation approach. Finally classification is to make the information retrieval easy. TF-IDF is an efficient information retrieval method which is performed for feature extraction [1].

2. RELATED WORK

TF-IDF weighting has been widely used for feature extraction. The idea is to determine words or terms that appear frequently in a document but that do not occur frequently in the entirety of each document collection. Many researchers have shown that TF-IDF is very effective in extracting features for scientific research, as in the work of [2]. Most text classification models use the TF-IDF method to weigh and extract the feature, but the TF-IDF method is based on word frequency and a predetermined threshold. TF-IDF was found to be effective in supporting subsequent machine learning efforts, such as k-NN. In this case, TF-IDF supports any adjustments that may be required in these machine learning tasks [3].

Features can be classified into positive or negative instances based on candidate words using a set of features. Features are not simply the most distinguishing or most specific words of a text. Other features, aside from the frequency in a corpus of texts, may therefore also be vital in feature selection. If feature extraction is processed as a supervised machine learning problem, then the integration of different kinds of features is straightforward. This method of feature extraction was presented [4] and [5]. The word co-occurrence distribution using a clustering approach to extract features for a single document without depending on a large corpus and found promising results. It is computationally efficient and performs reasonably well. Feature extraction has been processed as a supervised machine learning problem was determined by [6].

The supervised term weighting (STW), which is a term weighting methodology specific for IR applications related to supervised learning, such as TC and text filtering. Additionally, supervised term indexing leverages on the training data by weighing a term according to how different its distribution is in the positive and negative training examples was proposed by [7]. The automatic extraction and learning of key phrases from scientific articles written in English. The classification was performed by J48, which is an enhanced variant of C4.5 and a new technique (ConfWeight) to weigh features in the vector space model for text categorization by leveraging the classification task was conducted by [8]. TF-IDF is commonly used to show the text feature weight. However, it has certain problems: F-IDF cannot represent all the terms in the text and the importance degree and the difference among categories, as suggested by [9]. A new feature weighting method called TF-IDF-C, which shows the differences among categories by adding a new weight, was presented by [10].

3. PROPOSED SYSTEM

The proposed approach can be divided into three main stages such as: text preprocessing stage, term weighting techniques stage and dimensional reduction as shown in Figure 1.

3.1 TEXT PREPROCESSING

This stage is the most critical and complex, leading to the representation of each text document through the selection of a set of index terms. The major objective of text preprocessing is to obtain the key features or key terms from the text document dataset and to improve the relevance between word and document and that between word and category.

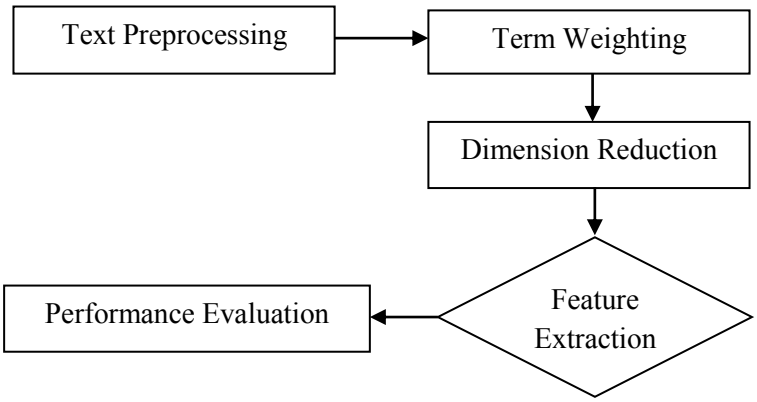

FIGURE 1. The stages of proposed system

After reading the input document, the text preprocessing stage divides the documents into features called tokens, words, terms, or attributes. These represent the text document in the form of a vector space, which consists of the features and their weights. The weights are obtained by the frequency of each feature in that text of the document. Following this, non-informative features, such as stop words, numbers, and special characters, are removed. Then, the remaining features are standardized by reducing them to their root using the stemming process. Despite the removal of non-informative features and the stemming process, the dimensionality of the feature space may still be too high. Thus, a threshold value (0.9) is applied to reduce the size of the feature space (between two words) for each input text document based on the frequency of each feature in that text document. These steps are used to prepare the text document.

3.1.1 TOKENIZATION

In this study, tokenization denotes not only the separation of strings into basic processing units but also the interpretation and grouping of isolated tokens to create higher-level tokens. Raw texts are preprocessed and segmented into textual units. The data must be processed in three operations: The first operation converts documents to word counts based on the bag of word (BOW). The second operation removes empty sequences; that is, this step consists of cleansing and filtering (e.g., whitespace collapsing and stripping extraneous control characters). In the final operation, each input text document is segmented into a list of features, which are also called tokens, words, terms, or attributes.

3.1.2 STOP WORDS ELIMINATION

Stop words are a list of commonly repeated features that emerge in each text document. Common features, such as conjunctions (e.g., “or,” “and,” “but,” and so on) and pronouns (e.g., “he,” “she,” “it,” and so on), need to be removed because they do not have any effect on or have minimal or no value in the categorization process. Thus, each stop word feature should be removed when it matches any feature in the stop word list. For the same reason, if the feature is a special character or a number, then it should be removed. To find the stop words, we can arrange our list of terms by frequency and select the high-frequency ones according to their lack

of semantic value. They should then be removed from text documents. Very rare words, such as those that occur only in matrix (m) or fewer documents (e.g., m= 6), can also be removed. The list of stop words used in this study was obtained from [11].

3.1.3 STEMMING

Stemming is the process of removing affixes (prefixes and suffixes) from features i.e. the process derived for reducing inflected (or sometimes derived) words to their stem. The stem need not to be identified to the original morphological root of the word and it is usually sufficiently related through words map to the similar stem. This process is used to reduce the number of features in the feature space and improve the performance of the clustering when the different forms of features are stemmed into a single feature. For example: (connect, connects, connected, and connecting) from the mentioned above example, the set of features is conflated into a single feature by removal of the different suffixes -s, -ed, -ing to get the single feature connect. This study applied standard Porter Stemming Algorithm for find the root words in the document.

3.2 TERM WEIGHTING

Term weighting can be as simple as binary representation or as detailed as a mix of terms and existing dataset. TF-IDF is the most widely known and used weighting method, and it remains comparable with novel methods. In TF-IDF term weighting, the text documents are characterized as transactions. In the text document vector, the correct weighting of a feature can enhance the performance of three different kinds of term weighting, namely, normal, global, and normalized, for a good comparison of different weighting configurations. Choosing the keyword for the feature extraction process is one of the main processes necessary to index the documents. Generally, it involves calculating the weight of a term i in a document j using Equation 1, is given by [12] as follows:

$$W_{ij} = N_{ij} G_j F_j \tag{1}$$

Where

TF refers to normal weighting (N_{ij}). It defines the weight i, which is the frequency or relative frequency of the word ti within a specific document, j.

IDF refers to global weighting (G_i). It defines the support of the word t_j with respect to the collection of documents.

F_j is the normalization factor for the term weights in document j, which is used to normalize the resulting document vectors. Thus, all the documents have the same modulus and can be compared without looking at the size of the text.

In this study, two different methods are explored, which are TF-IDF without and with logarithm, to weigh the terms in the term document matrices of the evaluation dataset. A word may appear in different topics, which leads to variability, and this variability can be defined as the degree of variations in documents according to the different types of category that include this word. A common practice to avoid this this variability or at least reduce the possible impacts resulting from it, is the normalization of the TF-IDF scores for each documents in the collection by using the Euclidean norm. The Euclidean norm specifies each vector in terms of rows in a

table. Thus, the Euclidean norm is identified as the magnitude. The calculations for TF-IDF are shown in Equation 2, as provided by [13], as follows:

$$(TF-IDF)_{ij} = TF_{ij} \times Log(IDF_j) \quad (2)$$

3.3 IMPROVEMENT OF TF-IDF METHOD

To address the issues of the current implementation of TF-IDF, this study explores the application of two different methods, that is, TF-IDF without and with logarithm, as shown in Equation 2.

$$(TF-IDF)_{hj} = Log(TF)_{ij} \times Log(IDF)_j \quad (3)$$

However, Equation 3 is used only when log (TF) ≥ 1. Otherwise, TF-IDF = 0 (see Equation (4)).

$$(TF-IDF)_{ij} = \begin{cases} \log(TF) \times \log(IDF) if \log(TF) \geq 1 \\ 0 otherwise \end{cases} \quad (4)$$

The TF-IDF method without logarithm is used for feature extraction. First, the keywords that represent the categories for the documents are obtained. Then, the document is transformed into numerical features by counting the number of occurrences (called term frequency) in the document and the word co-occurrences. The keyword is selected depending on a set threshold for each collection of documents. An important step is to choose the intended keywords that carry the meaning of the document and to reject others that do not [7]. Following this step, the keywords (or features) from the individual documents are combined.

4. DIMENSION REDUCTION

Dimension reduction is used in different applications, such as retrieval information, text classification, and data mining. The main goal of DR is to reduce high-dimensional data into a lower-dimensional subspace while keeping the essential features of the original data as much as possible. In this study, despite the removal of non-informative features, such as tokenization, stop word elimination, and stemming, the number of features in the feature space may still be too large. Thus, some features may hinder the classification task and may reduce the accuracy. Such features may be removed without affecting the classification performance. The classification learning may be affected because most machine-learning techniques cannot deal with a large number of features.

4.1 SINGULAR VALUE DECOMPOSITION

The major purpose of SVD is to reduce a dataset involving essentially values fewer values. In this study, the SVD of the TF-IDF matrix will be employed which is used in linear transformation approach. This matrix factorization leads to decompose a given n m matrix H into a correlated set of singular values and two orthogonal bases of singular vectors, as below:

$$H = USV^T \quad (5)$$

Where U and V are unitary matrices (i.e. $U^T = U^{-1}$ and $V^T = V^{-1}$) of dimensions n×n and m×m, S is an n×m diagonal matrix (i.e. $S_{ij} = 0$ if $i \neq j$), and T defines transposition. The diagonal elements of S are known as the singular values of H, and the columns of U and V are known as the ‘‘left’’ and ‘‘right’’ singular vectors of H, respectively. Observe that SVD does not represent a space reduction method per se.

Indeed, the n columns of U represents an orthogonal basis for the vector space spanned by the rows of H. Be reminded that the rows of a TF-IDF matrix correspond to vocabulary terms, so if H is a TF-IDF matrix, the columns of U would represent an orthogonal basis for the correlated document space. Similarly, the m columns of V represent an orthogonal basis for the vector space spanned by the columns of H. In this way, as the columns of a TF-IDF matrix correspond to documents, V would provide an orthogonal basis for the associated word space. These orthogonal bases are optimal values in the intelligence that they focus on the variability of the data in as few dimensions as possible. On the other hand, the first singular vector is supported with the direction in which the data shows its maximal variability, the second singular vector is aligned with the orthogonal direction (with respect to the first singular vector) in which the data shows maximal variability, the third singular vector is supported with the orthogonal direction (with respect to both the first and second singular vectors) in which the data shows maximal variability, and so on.

5. FEATURE EXTRACTION USING LOGARITHM TF-IDF METHOD

Given that the traditional TF-IDF (defined as TF × IDF) without logarithm considers keywords of low frequency to be significant and keywords of high frequency to be insignificant, this may not represent the usefulness or significance of certain keywords and may decrease the performance evaluation of classification. Thus, we include a logarithm before the traditional TF-IDF. However, given that log (IDF) may result in 0 if the IDF value is 1 (see Equation 4), a value of 1 is added to the equation (see Equation 6). This avoids the issue of division by 0 for the value of log (IDF).

$$(TF-IDF)_{ij} = \log(TF)_{ij} \times (1+\log(IDF)_j) \quad (6)$$

Thus, the weighting terms with the logarithm serves as an effective feature extraction method that reflects the weight of the keyword, which is in a particular category in the entire document collection.

Thereafter, the weights of the shortest feature are obtained by summing the frequencies of the conflated features.

The entire feature extraction process is listed as follows:

Step 1: Define an initial keyword set for each category of documents.

Step 2: Transform all the documents into numerical features, and label the keyword by category ID.

Step 3: Calculate the TF-IDF without and with logarithm.

Step 4: Perform feature extraction on the obtained numerical data, and label the document with the category ID.

Step 5: Calculate the initial accuracy based on the confusion matrix.

Step 6: Update the keyword set by adding and/or removing keywords in the current keyword set.

Step 7: Repeat from Step 2.

This process is continued until the desired features accuracy is obtained.

The steps of the dimension reduction are obtained by applying SVD while retaining the dimensionality largest singular values is listed as follows:

Step 1: Specify the dimensionality.

Step 2: Calculate the SVD.

6. EXPERIMENTAL RESULTS

6.1 DATASET

The dataset includes general topics that consist of 2,196 tweets from 12 general categories: politics (210), education (180), health (200), marketing (194), music (150), news and media (170), recreation and sports (190), computers and technology (192), pets (180), food (200), family (170), and others (160). These have also been filtered based on location (based on WOEID), which is either New York and Toronto.

6.2 PERFORMANCE EVALUATION

The comparison has to be implemented on the same algorithm and under the same environment. Hence, the algorithm must be implemented using the same technique and on the same platform. Thus, we used the same platform dataset but with varying sizes of the document. Figure 2 shows a performance comparison graph of the TF-IDF and Log (TF-IDF) methods in terms of the F1-measure. Blue the F1-measure of the TF-IDF method without logarithm, and red the F1-measure of the TF-IDF method with logarithm.

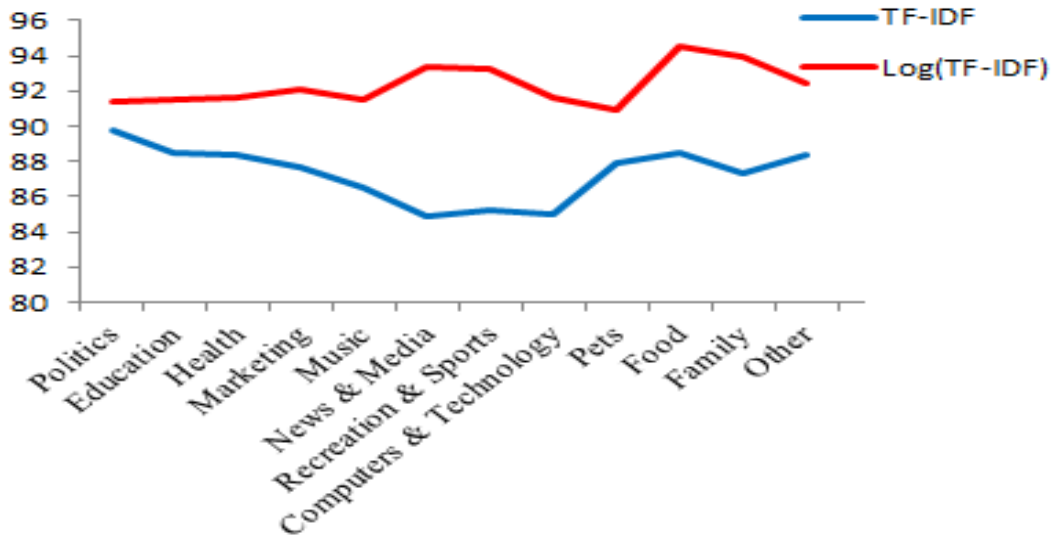

FIGURE 2. Comparison among the performance evaluation with respect to F1-measure on Twitter

The SVD approach is used to solve the "curse" of dimensionality through four cases for each dataset, that is, by reducing high-dimensional data into a lower-dimensional subspace. The main features of the original data are retained as much as possible. Different dimensions are investigated with an increase in the constant value (e.g., m = 50, 100, 150, and 200). The dimension is increased in a fixed

interval, that is, increments of 50, to determine the effect of the increase on the performance.

Figure 3 shows the four cases for the largest singular values of normalized pruned Log (TF-IDF) matrix. The decrease in processing time is due to the use of SVD, which is also used to reduce high-dimensional data to a lower-dimensional subspace [14]. Combining the two different Log (TF-IDF) with SVD methods enhances the performance of dimension reduction.

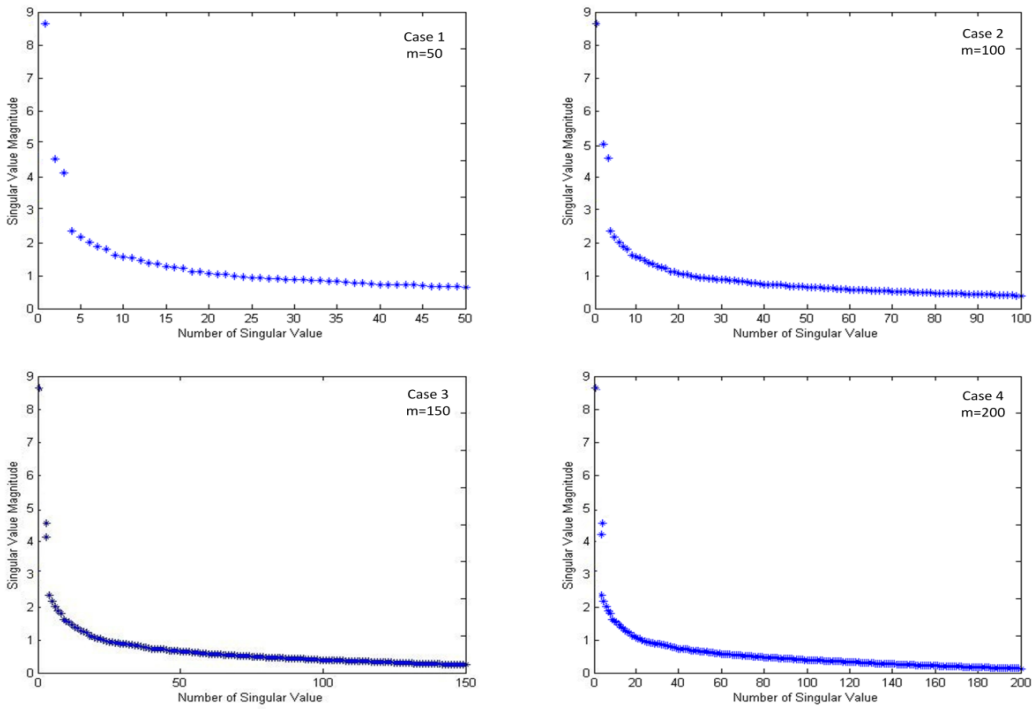

FIGURE 3. The four cases for the largest singular value magnitude using SVD approach on Twitter

As shown in Figure 3, when m = 200 (and above), the value of the singular value magnitude (SVM) not only becomes nearly constant but also decreases. Overall, the findings show that the average F1-measure increases with an increase in the number of dimensions (number of singular values) using SVD. That is, Log (TF-IDF) with SVD of Case 4 (m=200) outperforms the other cases as shown in Figure 4.

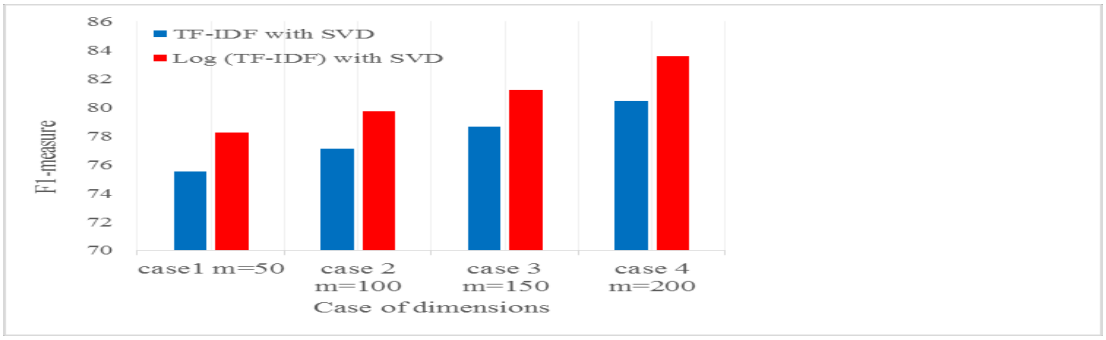

FIGURE 4. Comparison between the averages of F1-measure using TF-IDF and Log (TF-IDF) with SVD for each case on Twitter

7. CONCLUSION

In general, Log (TF-IDF) outperforms the traditional TF-IDF without logarithm according to the F1-measure. From experimental results, it is observed that the performance improves with increasing the number of features. Moreover, it showed that when m = 200 (and beyond), the value of SVM not only becomes nearly constant, but also achieves a lower value. The F1-measure increases with an increasing number of features as well as the Log (TF-IDF) with SVD has higher stability with increasing number of features.

REFERENCES

[1] Amato, G., F. Falchi, and C. Gennaro, On reducing the number of visual words in the bag-of-features representation, 2016.

[2] Wu, Y.-f.B., Li, Q., Bot, R.S., and Chen, X.: 'Domain-specific keyphrase extraction', in Editor (Ed.)^(Eds.): 'Book Domain-specific keyphrase extraction' (ACM, 2005, edn.), pp. 283-284, 2005.

[3] Turney, P. Coherent keyphrase extraction via web mining. The Eighteenth International Joint Conference on Artificial Intellegnce (IJCAI),9-15 August, Acapulco, Mexico, 434-439, 2003.

[4] Frank, E., Paynter, G. W., Witten, I. H., Gutwin, C., & Nevill-Manning, C. G. Domain-specific keyphrase extraction, In IJCAI, 668-673, 1999.

[5] Turney, P. D. Learning algorithms for keyphrase extraction. Information Retrieval, 2(4), 303-336, 2000.

[6] Matsuo, Y., and Ishizuka, M.: 'Keyword extraction from a single document using word co-occurrence statistical information', International Journal on Artificial Intelligence Tools, 13, (01), pp. 157-169, 2004.

[7] Debole, F., and Sebastiani, F.: 'Supervised term weighting for automated text categorization': 'Text mining and its applications', pp. 81-97, 2004.

[8] HaCohen-Kerner, Y., Gross, Z., and Masa, A.: 'Automatic extraction and learning of keyphrases from scientific articles', in Editor (Ed.)^(Eds.): 'Book Automatic extraction and learning of keyphrases from scientific articles', pp. 657-669, 2005.

[9] Soucy, P., and Mineau, G.W.: 'Beyond TFIDF weighting for text categorization in the vector space model', in Editor (Ed.)^(Eds.): 'Book Beyond TFIDF weighting for text categorization in the vector space model', pp. 1130-1135, 2005.

[10] Kuang, Q., and Xu, X.: 'An Improved Feature Weighting Method for Text Classification', AISS: Advances in Information Sciences and Service Sciences, 3, (7), pp. 340-346, 2011.

[11] Blanchard, A.: 'Understanding and customizing stopword lists for enhanced patent mapping', World Patent Information, 29, (4), pp. 308-316, 2007.

[12] Nassar, M. O., Kanaan, G., & Awad, H. A. Comparison between different global weighting schemes. The International Multi Conference of Engineers and Computer Scientists, 17-19 March, Hong Kong, China, Vol. 1, 2010.

[13] Salton, G., and Buckley, C.: 'Term-weighting approaches in automatic text retrieval', Information processing & management, 24, (5), pp. 513-523, 1988.

[14] [14] Biricik, G., Diri, B., & Sonmez, A. C. Abstract feature extraction for text classification. Turkish Journal of Electrical Engineering & Computer Sciences, 20(Sup. 1), 1137-1159, 2012.