Measuring the Impact of Using Different Tools on Classification System Results

Zainab A. Khalaf
Basrah University, College of Science, Department of Mathematics
E-mail: zainab.ali2004@gmail.com

Zainab M. Jawad
Basrah University, College of Education for Pure Sciences, Department of Computer Science
E-mail: z82m89asia@gmail.com

Abstract. A huge amount of textual data is available on the web. These data need to be classified under labels or classes to make the search more efficient and easier. Achieved by using automatic classification is used for this task. Many factors impact on the performance of the classifier system, such as the amount of using dataset, the data dispersion degree, preprocessing tools, feature extraction methods, terms weighting, and data reduction. So, researchers constantly compete to build a robust classifier with good performance. This study focuses on the effect of using different tools in preprocessing and term weighting stages. The experimental results applied on two different languages (Arabic and English languages). Also, the experimental results were compared with the recent related works.

1. Introduction
The amount of electronic textual data has rabidly increases, and this data is not organized based on categorizes. Consequently, searching and accessing the required data takes time. So, we require an automatic system capable of classifying data effectively and efficiently [1]. The text classification is the task that organizes the textual document (e.g. DOCX, HTML, TXT, PDF files) into classes based on predefined training documents. A lot of supervised machine learning was used as classifier like Naïve Bayes (NB), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Decision Trees, etc [2]. Many factors affects the classifier result, such as the amount of data training used, preprocessing tools (e.g. stemmer, part of speech tagging), feature extraction methods, and high dimensionality reduction factors. Generally, a large amount of data used to train the classifier for enhancing the classification result. However, this will cause a high dimensionality problem by increasing the number of irrelevant feature, which could negatively affect classification performance. Therefore, the feature reduction can be used to reduce the data dimensionality and improve the performance of the classification system [3].

The main objective of this paper to study the impact of tools on classification performance by using two feature reduction approaches. Also, the experimental results will compared with recent other research on the same datasets.
The organization of the paper is: the section (2) review the related work, in the section (3), the classification system will be explained, while the section (4) shows the proposed system method. The section (5) will discuss the experiments and the results, and then follows by the discussion of the results in section (6). Finally, the section (7) will summarize the conclusions and the future works.

2. Related Works

Researchers have improved the text classification either by reducing the features or enhancing the classification performance. Those tools could be the dataset, stemmers, number of stop words, feature reduction methods or the classification algorithm. The tools used in the classification will affect the classification positively or negatively; those tools could be the dataset, stemmers, number of stop words, feature reduction methods or the classification algorithm. Wan et al. [4] used bigram (which extracts two features that appear continuously), 2-termset (the two terms that occur in the same documents considered one feature) and SABigram (tokenizing the documents based on the stop words and the punctuations) in the preprocessing stage and applied the NB and SVM classifier on the 20-newsgroup dataset, the F1-score result was 64% and 82% respectively. also Dogan et al [5] used the 20-newsgroup and they used the tokenization based on white space and Porter Stemmer then applied the SVM, KNN and NN the F1-score results was 91%, 64% and 82% respectively.

The features affect the classification performance extensively because of the high dimensionality caused by the large vocabulary, especially in the big datasets. Therefore, many researchers try to reduce the features using different methods. Wan et al. [4] proposed a feature selection method called relevance category frequency (rcf) that considers the strength of the composite features and the ability to distinguish. They then combine the rcf with the Chi-square (Chi2). Furthermore, the researchers used three methods for feature extraction bigram, 2-termsets and the SABigram. The main disadvantage of the SABigram is that it is hard to find the entire combined features depending on the stop words and the punctuation tokenization; also, the rcf must combine with other feature reduction methods. Chandra [6] used three feature reduction methods the Chi2, Mutual Information (MI) and Term Frequency (TF). They applied these on three datasets using SVM and NB classifiers. They considered Chi2 as the best accuracy than the others. While Rehman et al [7] proposed a new feature selection method named Normalized Difference Measure (NDM) compared results with seven feature selection methods and seven datasets using two classifiers the SVM and NB. However, the NDM shows lower performance in the large datasets compared with the other feature reduction methods. On the other hand, Yang et al [8] compared the performance of six feature reduction methods: Information Gain (IG), Chi2, MI, Odd Ratio (OR), Gini Index (GINI) and Expected Cross Entropy (ECE) with their proposed method. During the preprocessing step, they depended on finding the probability of the feature in each class. If the probability was greater than the mean value, then the feature was selected. They applied the feature selection methods on the features extracted from the preprocessing for comparison and improved the performance of the original feature selection methods. However, they only used the Na’ive Bayes classifier to prove their method; they should try other classifiers for experimental verification. While, Mowafy et al [9] used the Chi2, IG, OR, Distinguishing features selector (DFS) and GINI feature selection and compared the results with the TF-IDF weighting, using the Multinomial NB and KNN. The result shows the preference for using Chi2, but they did not compare this method with other feature selection methods. In table 1 shows a related works with more details.

In this study we will compare the classification performance using TF-IDF and Chi2 methods. Moreover, we will compare our results with other studies to explain the different implementation using different functions.
# Table 1. The Related Works

| Reference No. | Year | Tools For Preprocessing | Feature Extraction Methods | Classification Model | No. of Features | Datasets Used For Evaluation | No. Of Classes | F1-Score | Accuracy |
|---------------|------|--------------------------|---------------------------|----------------------|----------------|-----------------------------|----------------|----------|----------|
| [4]           | 2019 | bigram 2-termset SABigram | Chia, ref                 | SVM, NB              | 5500           | 20-newsgroup (19997)       | 20             | 82%      | 94%      |
|               |      |                          |                           |                      | 6705           | Reuters-21578              | 10             | 93%      | 56%      |
| [6]           | 2019 | Chia, MI, TF              | SVM, NB                   | 20%                  | 20-newsgroup (18846) | 20             | 92%            | 92%      | 86%      | 87%      |
|               |      |                          |                           |                      |                | Reuters-21578              | 10             |          |          |
| [5]           | 2019 | Tokenization Normalization | TF-IDF                    | SVM, KNN             | 25000          | 20-newsgroup (19997)       | 20             | 91%      | 64%      |
|               |      | (Porter Stemming)         |                           |                      | 8000           | Reuters-21578              | 10             | 82%      | 77%      |
|               |      |                          |                           |                      |                | 100                      |                |          | 81%      |
| [8]           | 2019 | Tokenization Normalization | IG, Chia, MI, OR,        | NB                   | 2000           | 20-newsgroup (19997)       | 20             | 78%      | 84%      |
|               |      | (Porter Stemming)         | GINI, ECE,                |                      | 2500           | Reuters-21578              | 10             |          |          |
|               |      |                          |                           |                      |                | 100                      |                |          |          |
| [10]          | 2019 | TF more than 2            | SVM, NB                   | 76%                  | 43553          | 20-newsgroup (19997)       | 20             | 61%      | 85%      |
|               |      |                          |                           |                      | 50570          | 20-newsgroup (18828)       | 20             | 75%      | 86%      |
| [11]          | 2018 | Tokenization Normalization | TF-IDF                    | NB + KNN             | 4856           | 20-newsgroup (19997)       | 20             | 87%      | 86%      |
|               |      | (Porter Stemming)         |                           |                      | 10913          | Reuters-21578              | 50             |          |          |
|               |      |                          |                           | SVM+uni-gram        |                |                            |                |          | 88%      |
| [7]           | 2017 | IG, Chia, OR, DFS, GINI  | SVM, NB                   | 1500                 | 20-newsgroup (19997) | 20             | 75%            | 73%      |          |          |

## 3. Text Classification System

The system that assigns a group of textual documents into one or more predefined classes based on their subjects automatically called text classification system. Let $D$ is set of learning documents that belong to different classes, where $D = \{d_1, \ldots, d_n\}$ with predefined classes $C = \{c_1, \ldots, c_m\}$, $n$ and $m$ are the number of documents and predefined classes respectively. And, suppose $x$ is a new document, so the classification system tries to predicate class ($c_x$) to document $x$ based on the most likely similarity classes in $C$, that is $c_x$ in $C$.

The text classification system consists of five main steps, as shown in figure (1). Data Collection is a set of unstructured documents collected from different resources used to learning the classifier model. Preprocessing is the primary step used to prepare the unstructured documents for subsequent processing. Representation and Dimension Reduction are the processes aiming to represent the weighting features and reduce the high dimensionality of the data. Classification Algorithms are the technique used to predict the class label of the new document based on the predefined pattern. Performance Evaluation is a process used to measure classifier performance.
4. The Proposed System

The text classification is automatically labelling the new documents based on the training of predefined labelled documents. The first step of any classification system is preparing the input documents for processing. Then, the critical features are extracted and passed as input for the classifier. Finally, we evaluate the classifier results in the next stage, explained in the following subsections:

4.1. Preprocessing Stage

The preprocessing aims to prepare the data collection for processing [12][13]. Preprocessing includes many subtasks:

- **Tokenization**: is cutting the documents into tokens by using the space for separation.
- **Normalization**: a subtask used to exclude the white space, special symbols and characters, as well as unify the shape of letters, such as convert uppercase letters to lowercase letters for English language, or unify the shape of some of the Arabic letters.
- **Stop-words**: the list of the words that frequently appear in the language like conjunctions and prepositions. These words are removed to reduce the space model of classification.
- **Stemming and lemmatization**: exclude suffixes and prefixes from words to reduce the word forms. The current study, uses Stanford corenlp lemmatizer for lemmatization. While, porter stemmer and ArabicLightStemme are used for English and Arabic, respectively. The remaining terms obtained from the preprocessing step
called features are used to create the vector space model (VSM). The VSM dimension is \( n \times m \), where \( n \) is the number of classes, and \( m \) is the number of features. Each cell in VSM represents the weight of the feature \((f_j)\) in the class \((c_i)\) [14][7].

4.2. Dimension Reduction

The performance of text classification depends on the amount of trained dataset. Moreover, a large dataset causes high dimensionality. Therefore, dimension reduction is needed to decrease the computations budget without impacting the classifier performance. For this reason, the important features can be extracted from the large dataset and thus exclude irrelevant features. TF-IDF and Chi2 consider one of the effective methods to weighting and extract features by selecting the best features weight.

(i) TF-IDF: is calculated by finding the TF and IDF as the following equation [15]:

\[
TF = \log_2(tf_{ij})
\]

\( tf_{ij} \) is the term frequency of term \( i \) in document \( j \)

\[
IDF = \log_2 \left( \frac{N}{n_j} \right)
\]

\( N \) is the number of documents in the collection, \( n_j \) the number of documents containing the term \( j \).

And the TF-IDF will calculated as the following:

\[
TF - IDF = TF \ast IDF
\]

(ii) Chi2: Chi-square is one of the main feature reduction methods. The basic idea of Chi2 is to compare the distribution of actual data and the distribution of expected data. Suppose \( C \) is the classes \( c_1, c_2, \ldots, c_n \), and \( F \) is the feature \( f_1, f_2, \ldots, f_r \). The data documents will described by \( C \) and \( F \). Let \((C_i, F_j)\) denote the event that feature \( C \) takes on value \( c_i \) and feature \( F \) takes on value \( f_j \). The equation will calculate Chi2 [16][17]:

\[
Chi2 = \sum_{i=1}^{r} \sum_{j=1}^{c} \frac{(n_{ij} - e_{ij})^2}{e_{ij}}
\]

Where \( n_i \): the expected frequency (actual), \( e_i \): the frequency that gained (observed).

The \( e_{ij} \) calculated by:

\[
e_{ij} = \frac{(n_i n_j)}{n}
\]

\( n_i \): the number of the documents that belongs to the class \( i \). \( n_j \): the number of documents that have the feature \( j \).

4.3. Classification Algorithms

The classifiers used in this study are Naïve Bayes (NB), Support Vector Machine (SVM) and K-Nearest Neighbors (KNN).

- Naïve Bayes (NB): NB is a simple and efficient probabilistic algorithm based on the distribution probability between the documents and the classes. The training phase obtains a set of parameters, which are used in the testing phase to predict the probability of the document belonging to the class [12][18].
Support Vector Machine (SVM): SVM is a stable and effective statistical classifier. The main idea of the SVM is using the training data to find the hyperplane with the highest Margin, where the margin is the largest distance between the nearest points on both sides of the hyperplane. The hyperplane used in the testing phase to decide the class of the new document [19][20][21].

K-Nearest Neighbors (KNN): KNN is a simple and easy to understand algorithm. The main idea is to find the distance between the new document and the training data documents and vote for the k nearest (smallest distance) documents. There are different measures used like Euclidean distance, cosine measure, etc. The similarity score of each nearest neighbor document to the test document is used as the weight of the classes of the neighbor document [22].

4.4. Performance Evaluation

The main performance measures widely used in text classification are the Precision, Recall and F1-score. Precision (P) is indicated as the ratio between the number of properly classified documents (Correct Classes) to all documents that can be automatically recognized by the class (Predicated Classes). The recall (R) is defined as the ratio between the number of properly classified documents (Correct Classes) to all documents belonging to that class (Actual Classes). The F1-score indicates the harmonic average between precision and recall [23][24].

\[
\text{Precision} = \frac{\text{CorrectClasses}}{\text{PredicatedClasses}} \tag{6}
\]

\[
\text{Recall} = \frac{\text{CorrectClasses}}{\text{ActualClasses}} \tag{7}
\]

\[
F1\_\text{score} = \frac{2 \ast (\text{Recall} \ast \text{Precision})}{(\text{Recall} + \text{Precision})} \tag{8}
\]

5. Experimental Results & Discussion

In this section, the performance of the proposed system will be evaluated. TF-IDF and Chi2 are used for feature selection; different percentages are used for dimension reduction by selecting the best features. Also, we compare many related works using the same datasets.

5.1. Datasets

Two languages are used in the current study: English and Arabic. For the English language, 20-newsgroups [25] and Reuters-21578 [26] are used; for the Arabic language, Watan-2004 [27] and Khalaf-2018 are used. The 20-newsgroup contains twenty classes with 19997 documents, and the top seven classes from Reuters-21578 data in 10806 documents are used. The Watan-2004 dataset contains around 20291 documents in six classes, while Khalaf-2018, which is collected manually by [28] from the BBC and CNN online websites containing of 2750 Arabic news belongs to six classes. All these datasets are divided into 75% documents for training and 25% for testing.
5.2. The Classification System Results

After applying the preprocessing stage used to create the VSM, the features extracted from this stage are 86151, 18431, 78130 and 31980 for 20-newsgroup, Reuters-21578, Watan-2004 and Khalaf-2018, respectively. Three classification systems (NB, SVM and KNN) are applied with the parameters for NB is the default multinomialNB, linear SVM and KNN using Euclidian distance when K=5.

The experiment examined six percentages of reduction (80%, 70%, 60%, 55%, 50%) from the original features by applying TF-IDF and Chi2. The best result was with the 55% reduction. The number of feature reduced to 47383, 10137, 42971 and 11193 for 20-newsgroup, Reuters-21578, Watan-2994 and Khalaf-2018, respectively. The (table 2) shows the results of the classification system. The F1-score was enhanced for NB and SVM classification system, The KNN performs the worst. The reason is because of the neighbor features may introduce more noisy information instead of useful information.

Table 2. The Evaluation Of Current Study

| No. Of Features | 20-Newsgroup | Reuters-21578 | Watan-2004 | Khalaf-2018 |
|-----------------|--------------|---------------|------------|-------------|
| Before Reduction| 86151        | 18431         | 78130      | 31980       |
| After Reduction | 47383        | 10137         | 42971      | 11193       |
| TF-IDF          | 91%          | 1%            | 88%        | 82%         |
| Chi2            | 9%           | 1%            | 82%        | 88%         |
| TF-IDF          | 95%          | 5%            | 93%        | 94%         |
| Chi2            | 9%           | 1%            | 94%        | 95%         |
| TF-IDF          | 80%          | 60%           | 83%        | 79%         |
| Chi2            | 86%          | 91%           | 93%        | 86%         |

6. Discussion

This study used three classifiers to classify four datasets, then applied two feature reduction methods: TF-IDF and Chi2. Feature selection with different rank ratios (50%-80%) are evaluated in this study is to find the best ratio for evaluation. The best rank ratio obtained using the TF-IDF and Chi2 is 55%. When comparing the TF-IDF and Chi2 feature reduction methods, the TF-IDF shows a better enhancement in the English datasets, while the Chi2 was the best for Arabic datasets.

In the other hand, the (table 3) shows other researchers’ results on the using datasets. The F1-score results of the current study show a better performance using the four datasets over the results of the other studies. Also, the table indicates the difference in the preprocessing tools and how the stemmers and stop words effect the classification performance.

For the 20-newsgroup dataset, Dogan et al [5] split the data into 50:50 for training and testing. Porter stemmer and TF-IDF reduction ( the number of features after reduction was 25000) were used, and then the obtained features were passed to the SVM, KNN and Neural Networks (NN) classifiers. Despite their number of the features being less than the current study, the accuracy result of the current study is better. Furthermore, Wan et al [4] proposed
a feature selection method called Chi2 and rcf. The accuracy results of current study is better than their results. As mentioned in section 2, they used three methods of feature extraction. However, some disadvantages could affect the classification performance. Also, the current study showed a better accuracy performance than the results in [4] and [5].

For Reuters-21578, the researchers Unnikrishnan et al [29] also used the top seven classes from Reuters-21578. In the preprocessing, they used a Porter Stemmer algorithm with removing the words that occur at least in all the classes but one. Furthermore, the SVM and KNN were applied to compare with their proposed system, while the current study applied the Stanford lemmatizer and Porter stemmer, and the entire feature extracted was used and achieved better performance.

In the Watan-2004 dataset, the researchers Sabbah et al [30] selected 9,000 document from the original dataset and applied the Arabic Light Stemmer with using Chi2, IG and proposed SVM-FRM for feature selection and got 93% for SVM classifier. The current study, used all the documents of the datasets and got a better result of 94% for SVM classifier.

For Khalaf-2018, the researcher Khalaf et al [28] used the Khoja stemmer from safar library and the stop list containing (10,471) words in the preprocessing stage. Their SVM result was 91% using 50% from the features. While the current study used the same dataset by applying Stanford core nlp and ArabicLightStemmer, the stop list contained (15,847) words in the preprocessing stage. The results of the current study were 91% with less number of features.

| Reference No. | Year | Tools For Preprocessing | Feature Extraction Methods | Classification Model | No. of Features | Datasets Used For Evaluation | No. Of Classes | F1-Score |
|---------------|------|-------------------------|---------------------------|---------------------|-----------------|-----------------------------|----------------|---------|
| [5] 2019      |      | Tokenization Normalization stop list stemming (Porter Stemmer) | TF-IDF                   | SVM, KNN            | 25000           | 20-newsgroup (19997)        | 20             | 91%, 64% |
| [29] 2019     |      | Porter stemmer and remove the words that occur in all the classes | TF                        | SVM, KNN            | 1500            | Reuters-21578               | 7              | 86%, 85%, 78% |
| [30] 2011     |      | Arabic Light stemmer | Chi1, IG                  | SVM                | 5000            | Watan-2004 (9000 documents) | 6              | 93%     |
| [28] 2018     |      | khoja stemmer (10471) stop words. | N-gram | SVM             | 12474           | Khalaf-2018               | 6              | 91%     |
| Current study |      | Porter stemmer Stanford Lemmatizer remove the words that occur in all classes (797) stop words | TF-IDF                   | NB, SVM, KNN       | 47383           | 20-newsgroup               | 20             | 91%, 95%, 80% |
|              |      | ArabicLightStemmer Stanford Lemmatizer remove the words that occur in all classes (15847) stop words | CHI2                     |                   | 10137           | Reuters-21578              | 7              | 88%, 93%, 83% |
|              |      |                          |                           |                   | 42971           | Watan-2004                 | 6              | 88%, 94%, 93% |
|              |      |                          |                           |                   | 11193           | Khalaf-2018                | 6              | 73%, 91%, 91% |

7. Conclusions and Future works
The most important issue in the text classification is efficiency and the reliability. Therefore, the researchers who study text classification try to produce the tools and functions that perform efficiently. In this study, we compared the performance of two feature selection methods TF-IDF and Chi2 by finding the best percentage of reduction from the original feature number. The best F1-score result was in the 55% feature reduction in both the TF-IDF and Chi2. However, the English datasets show better TF-DF results than the Chi2, unlike the Arabic dataset, which found that Chi2 was best. Also, we compared results with other recent research findings to illustrate how the tool affects classification performance. Thus, it is important to find the best tools when building a classification system.

Future research could improve the classification system using other feature reduction methods, different classifiers or by enhancing the functions used in the text classification system (e.g. stemmers, feature reduction or classifiers).

References
[1] Ge Lihaor & Moh Teng-Sheng, (2018), “Improving Text Classification With Word Embedding,” Proceedings - 2017 IEEE International Conference on Big Data, pp. 1796 – 1805.
[2] Bai Ruijiang & Wang Xiaoyue & and Liao Junhua, (2010), “Extract Semantic Information From Wordnet To Improve”, Advances in Computer Science and Information Technology conference, Springer-Verlag Berlin Heidelberg New York, Vol. 6059, pp. 409 – 420.
[3] Kadhim Ammar, (2019), “Survey On Supervised Machine Learning Techniques For Automatic Text Classification”, Springer Artificial Intelligence Review, Vol. 52, pp. 273 – 292.
[4] Wan Chuan, Wang Yuling, Liu Yaoze, ji Jinchao & Feng, Guozhong, (2019), “Composite Feature Extraction And Selection For Text Classification”, IEEE Access, Vol. 7, pp.1-1.
[5] Dogan Turgut & Uysal Alper Kursat, (2019), “Improved inverse gravity moment term weighting for text classification,” Elsevier, Expert Systems with Applications, Vol. 130, pp. 45 – 59.
[6] Chandra Andreas, (2019), “Comparison Of Feature Selection For Imbalance Text Datasets,” International Conference on Information Management and Technology, Vol. 1, No. August, pp. 68 – 72.
[7] Rehman Abdur, Javed Kashif & Babri Haroon, (2017), “Feature Selection Based On A Normalized Difference Measure For Text Classification”, Elsevier Information Processing and Management, Vol. 53, No. 2, pp. 473 – 489.
[8] Yang Jieming, Lu Yixin & Liu Zhijing, (2019), “An Improved Strategy of the Feature Selection Algorithm for the Text Categorization,” International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, No. 2, pp. 3 – 7.
[9] Mowafy M, Rezk A & El-bakry HM, (2018), “iMedPub Journals An Efficient Classification Model for Unstructured Text Document Keywords : Introduction”, American Journal of Computer Science and Information Technology, Vol. 6, No.1:16 pp.1-10.
[10] Kilime Zeynepi & Akyokus Selim, (2018), “Deep Learning- And Word Embedding-Based Heterogeneous Classifier Ensembles For Text Classification”, Wiley Hidawi Complexity, Vol. 2018, pp. 1 – 10.
[11] Panda Mrutyunjaya, (2018), “Developing an Efficient Text Pre-Processing Method with Sparse Generative Naïve Bayes for Text Mining,” International Journal of Modern Education and Computer Science, Vol. 10, No. 9, pp. 11 – 19.
[12] Kobavashi Vladimir & Mol Stefan & Berkers Hannah & Kismihök Gabor & Hartog Den, (2018), “Text Classification For Organizational Researchers: A Tutorial”, Organizational Research Methods, Vol. 21, No. 3, pp. 766 – 799.
[13] Li Qi-na and Li Ting-hui, (2020), “Research on the application of Naïve Bayes and Support Vector Machine algorithm on exercises Classification,” Journal of Physics: Conference Series, Vol. 1437, No. 1.
[14] Anandarajan Murugan, Hill Chelsey & Nolan Thomas, (2019), “Classification Analysis: Machine Learning Applied To Text. In: Practical Text Analytics”, Advances in Analytics and Data Science, Vol 2. Springer, Cham.
[15] Anantharaman Adita, Jadiya Arpit, Tulasi Chandana, Siri Saj , Nvs Adikar Bharath & Mohan Biju, (2019), “Performance Evaluation of Topic Modeling Algorithms for Text Classification,” , IEEE, 3rd International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, pp. 704 – 708.
[16] Han Jiawei, and Kamber Micheline, (2006), “Data Mining: Concepts and Techniques”, second edition, Amsterdam: Elsevier/Morgan Kaufmann.
[17] Daniah Muhammad, Efendi Syahril & Sembiring Rahmat Widia , (2019), “The Analysis of Attribution Reduction of K-Nearest Neighbor (KNN) Algorithm by Using Chi-Square”, Journal of Physics: Conference Series. Vol.1424 012004.
[18] Patra Anuradha & Singh Divakar, (2016), “A Survey Report On Text Classification With Different Term Weighing Methods And Comparison Between Classification Algorithms”, International Journal of Computer Applications, Vol. 75, No. 7.

[19] Akay Mehmet Fatih, (2017), “Support Vector Machines Combined With Feature Selection For Diabetes Diagnosis,” Elsevier, Journal of Electrical and Electronics Engineering, Vol. 17, No. 2, pp. 3219 – 3225.

[20] Karamizadeh Sasan, Abdullah Shahidan & Halimi Mehran, (2014), “Advantage And Drawback Of Support Vector Machine Functionality”, IEEE, 2014 International Conference on Computer, Communication and Control Technology in Malaysia, pp. 63 – 65.

[21] Rustam Z., Nadhifa F. & Acar M, (2018), ” Comparison of SVM and FSVM for predicting bank failures using chi-square feature selection”, Journal of Physics: Conference Series. Vol.1108, 012115

[22] Uguz Harun, (2011), “A Two-Stage Feature Selection Method For Text Categorization By Using Information Gain, Principal Component Analysis And Genetic Algorithm” Elsevier Knowledge-Based System, Vol. 24, No. 7, pp. 1024 – 1032.

[23] Wang You-wei & Feng Li-Zhou, (2018) “A New Feature Selection Method For Handling Redundant Information In Text Classification,”, Frontiers of Information Technology & Electronic Engineering, Vol. 19, No. 2, pp. 221 – 234.

[24] Yin Chunyong & Xi Jinwen, (2017), “Maximum Entropy Model For Mobile Text Classification In Cloud Computing Using Improved Information Gain Algorithm,”, Springer, Multimedia Tools and Applications, Vol. 76, No. 16, pp. 16875 – 16891.

[25] 20Newsgroups. http://qwone.com/ jason/20Newsgroups/.

[26] Reuters, http://konect.cc/networks/gottron-reuters/.

[27] Watan.pdf: https://sourceforge.net/projects/arabiccorpus/.

[28] Khalaf Zainab A. & Hassan Khadeija A., (2018), “Filtering Approach And System Combination For Arabic News Classification,” Journal of Theoretical and Applied Information Technology, Vol. 96, No. 14, pp. 4491 – 4501.

[29] Unnikrishnan P., Govindan V. K., & Madhu Kumar, (2019), “Enhanced sparse representation classifier for text classification,” Expert Systems with Applications, Vol. 129, pp. 260 – 272.

[30] Sabbah Thabit, Ayyash Mosab & Ashraf Mahmood, (2018), “Hybrid support vector machine based feature selection method for text classification,” , International Arab Journal of Information Technology, Vol. 15, No. 3A Special Issue, pp. 599 – 609.