ACIT'2017

The International Arab Conference on Information Technology

Yassmine Hammamet, Tunisia

December 22-24, 2017

Support Vector Machine based Feature Selection Method for Text Classification

Thabit Sabbah, Mosab Ayyash and Mahmood Ashraf
Faculty of Technology and Applied Sciences, Al-Quds Open University, Ramallah, Palestine
Department of Computer Science, Federal Urdu University of Arts, Science & Technology, Islamabad, Pakistan

***Abstract*—Automatic text classification is one of the most effective tools used to sort out the increasing amount of textual content available online. High dimensionality remains one of the major obstacles observed in the text classification field in spite of the fact that there have been statistical methods available to face this issue. Still, none of them has proved to be effective enough to solve this problem. This paper proposes a feature ranking and selection method based on the Support Vector Machine (SVM) learning algorithm, also known as (SVM-FRM). This method assumes that weights given by the SVM learning algorithm to different features in feature space indicate the significance of these features. As such, the feature selection process can be established based on the referred to weights. The researchers tested the SVM proposed method using three text classification public datasets. Then, they compared the results to those of other statistical feature selection methods currently used for this purpose. In the light of this comparison, applying the proposed SVM-FRM method for text classification has proved to have a superior F-measure and accuracy performances than the rest of other methods applied for this purpose, when tested on balanced datasets, in spite of its size and the high competing performances on an unbalanced dataset.**

Keywords*— *Feature ranking; text classification; feature selection; SVM; dimensionality reduction;

I. Introduction

The rapidly increasing amount of online textual content makes machine learning- based text categorization or classification one of the most effective solutions for knowledge management and information organization. Text Classification (TC) refers to the process of predicting the category of a document and associating it to predefined classes or categories [1]. The most fundamental step in TC is text representation and feature selection which enables the classification algorithms to deal with textual content [2] and reduce the size of feature space.

For many decades, Vector Space Model (VSM) has proved to be an effective representation method that enables different classification algorithms to process a collection of various documents. VSM is applicable in several domains such as summarization [3], recommender systems [4], and text categorization [5]. VSM model represents documents as vectors of weighted features, where features can be of different types. It also provides many weighting methods [6]. In spite of the available features' types and weighting methods, the huge dimensionality of feature space is a major problem that should be reduced to decrease the computational complexity, increase the classification algorithms performance, and reduce the resources required for data processing [7].

Feature Selection (FS) methods are part of dimensionality reduction methods that aim at downsizing the dimensionality of feature space. When applying an FS method, the most informative features are selected while the less important and uninformative features are eliminated, based on the assumption that removing such features will not significantly affect the quality of the classification [8]. However, selecting the most informative features involves the process of weighting and ranking all features in the feature space. In general, this process is based on the statistical analysis of feature space that analyzes the intrinsic characteristics of the document [9] or the corpus [6].

In TC domain, many methods for feature weighting and ranking have been used frequently such as Document Frequency – Inverse Document Frequency (TFIDF) [10, 11], Term Frequency (TF) [10], Term Frequency – Relevance Frequency (TFRF) [2, 12], Document Frequency (DF) [13], Chi-square (CHI) [14], Entropy [15], Inverse Document Frequency (IDF) [16], Information Gain (IG) [13], Gini Index (GI) [17], Improved Gini Index (GINI) [18], and Correlation [19]. Based on the literature available in this area, other less common methods have been proposed for feature weighting and ranking such as Balanced Term Weighting Scheme (BTWS) [20], Term Variance (TV) [13], Glasgow [21, 22], Odds Ratio (OR) [18], Mutual Information (MI) [23], Term Strength (TS) [24], and many other modified schemes[25].

However, the mentioned above ranking methods depend on the statistical analysis of feature space. As such, this paper proposes a feature ranking and selection method in which the weights are assigned according to a learning algorithm. The proposed method is based on the assumption that the more informative and important the features are, the higher the weights they are assigned by the learning algorithm. As such, the feature selection based on these weights will eventually lead to higher classification performance.

This paper consists of the following sections in addition to this introductory section; Section II presents some of the studies and researches related to the TC domain. It explains the basics of the Support Vector Machine (SVM) learning algorithm which will be utilized in the method proposed in this paper. Section III describes the proposed Support Vector Machine based Feature Ranking Method (SVM-FRM). Section IV highlights the used datasets and the conducted experiments. Section V presents the obtained results along with a discussion on the major findings. Section VI provides the conclusion on this paper and suggests headlines for future studies to be conducted by the research team.

II. Related Works

This section presents major concepts of feature ranking based on Information Gain (IG), Correlation, Chi-square, and the Support Vector Machine (SVM) learning algorithm.

A. *Information Gain (IG)*

Information Gain (IG) is among the most commonly applied feature selection methods [26]. It is a statistic that measures the goodness of an attribute (i.e. feature). As previously referred to, feature reduction methods aim at determining and applying the most useful attributes for distinguishing the different classes of a given feature space. Therefore, IG measure can indicate how important each of the attributes is, by calculating the weight (relevance) of an attribute in terms of the class attributes. The higher the weight of an attribute, the more distinguished it is considered to be.

The IG of a feature *f* is defined as the information gained by doing the split of the feature space based on that particular feature, which is mathematically expressed as follows [27, 28]:

$$\begin{aligned} IG(f) = & -\sum_{i=1}^{m} P_r(c_i)\log P_r(c_i) \\ & + P_r(f)\sum_{i=1}^{m} P_r(c_i \mid f)\log P_r(c_i \mid f) \\ & + P_r(\overline{f})\sum_{i=1}^{m} P_r(c_i \mid \overline{f})\log P_r(c_i \mid \overline{f}) \end{aligned} \tag{1}$$

where *m* is the number of categories, *P(c$_i$)* is the probability of category *c$_i$*, *P$_r$(f)* and *P$_r$($\overline{f}$)* are the probabilities of occurrence and nonappearance of feature *f*, *P(c$_i$|f)* and *P(c$_i$|$\overline{f}$)* are the conditional probabilities of category *c$_i$* considering presence and absence of feature *f*, respectively.

Although IG is a good measure for an attribute's relevance, it has lower performance when it is applied to attributes that can take a large number of distinct values. More details on IG can be found in [29].

B. *Correlation*

Correlation statistic is used to measure the linear association (correlation) between two attributes (i.e. features), where attributes of higher correlation weight are considered to be more relevant. A correlation is defined as a number ranging from -1 to +1 that represents the degree of association between two attributes (let these attributes be X and Y). A positive association between X and Y is represented by a positive value for the correlation while a negative correlation value implies an inverse or negative association [30]. The correlation of two attribute vectors X and Y is defined as follows:

$$Correlation(X,Y) = \frac{\sum_{i=1}^{n}(X(i)-\overline{X}).(Y(i)-\overline{Y})}{(n-1).\sigma(X).\sigma(Y)} \tag{2}$$

Where *n* is the number of samples (i.e. document). $\overline{X}$, σ(X) and $\overline{Y}$, σ(Y) are the means and standard deviations of X and Y, respectively.

However, using correlation for feature selection involves finding a subset of features in which the features are correlated as less as possible among each other. Besides, each of them, i.e. the features, has to be correlated with classes vector as much as possible. Usually, correlation based feature selection is based on heuristic search strategies to find the appropriate feature subset in a reasonable period of time [19].

C. *Chi-Square*

Similar to the IG and Correlation, chi-square is a nonparametric statistical technique used to compute the lack of independence between the distributions of observed frequencies and the theoretically expected frequencies [27], where the higher the weight of an attribute, the more relevant it is. In general, Chi-square statistics use nominal data. In the TC domain, however, it uses feature's frequencies instead of using means and variances. The value of the chi-square statistic is given by:

$$\chi^2 = Sigma\left[\frac{(O-E)^2}{E}\right] \tag{3}$$

Where chi-square statistic is noted as χ^2, *O* is the observed frequency and *E* is the expected frequency. More details on chi-square in TC domain is provided in [31].

D. *Support Vector Machine*

Support Vector Machine (SVM) is one of the most popular and effective supervised learning algorithms [28]. It depends on learning from a training set to find a hyper-plane that can separate the cases of binary classes [32]. The hyper-plane is located at the point in the hyper-space that maximizes the distance between the support vectors which are the closest positive and negative samples. Two components play a vital role in Linear SVM; one is a weight vector Ẅ which is perpendicular to the hyper-plane. The other one is the bias ḃ which is the offset of hyper-plane from the origin. The class of an unlabeled example Ẍ is determined by calculating the value f(ẍ), where f(ẍ) = ẄẌ + ḃ. If the computed value of f(ẍ) is greater than or equals zero, the example is classified as positive. Otherwise, it is classified as negative.

SVM algorithm has many advantages that make it preferable among other classification tools. Among which is the ability to handle extremely large feature spaces besides the well-handling of high dimensional feature vectors and redundant features which are the features that can be predicted from others. SVM has also been proved, in various domains including text classification, to be among the best performing machine learning approaches [33]. SVM is an effective binary classifier that has been utilized by many existing projects as text classifier. It can be applied for multi label classification. For example, [34] presents a comprehensive empirical comparison study in which many different SVM algorithms were tested on various publicly text classification datasets.

In this paper, the research team utilizes the weight vector $\ddot{W}$ based on the assumption that for each w_i represents the contribution and importance of feature f_i to the separation hyper-plane.

III. Proposed Support Vector Machine based Feature Ranking and Selection Method (SVM-FRM)

This paper presents a Support Vector Machine based Feature Ranking Method (SVM-FRM). This method utilizes the SVM learning algorithm in order to assign weight values to the features in the feature space. Then, the referred to weights are used as ranking criteria to select the top features for classification. The proposed method is part of the general text classification approach that consists of many steps. The major three steps are summarized as follows. The TFIDF weighting method is used for the text representation. Then, the SVM-FRM is applied. Finally, the top K features are used in the classification process. Fig. 1 shows the steps of the general approach, followed by the detailed description of the mentioned steps.

A. *Step 1: Preprocessing*

This step includes the application of case transformation, filtering, stop-words removal and stemming methods. Filtering includes eliminating the non-words tokens from the text such as numbers, Latin words, and Html tags. In this research, filtering also removes from the text the words of less than four or more than fifteen characters in length. Stop-words removal process usually removes the meaningless tokens from the text. The default stop-words list for the Arabic language included in RapidMiner Studio 7.5 was the one referred to for the purposes of this research. Stemming is the process of reducing inflected words to their word stem. It should be said that the simple form of stemming is to treat related words as synonyms of the same stem when even this stem may not be a valid root. The Arabic Light stemmer is referred to in this step.

B. *Step 2: Text Representation*

In this step, the corpus is represented based on the Vector Space Model (VSM) in which the different term weighting formula can be considered for document vector creation [25]. This paper uses the Term frequency-Inverse Document Frequency (TF-IDF) weighting formula because of its popularity and efficiency in the domain of text classification. The result of this step is the weighting matrix in which each row represents a document while each column represents one feature. In this research, features are considered as the unigram token basis (i.e. each single word is considered as one feature).

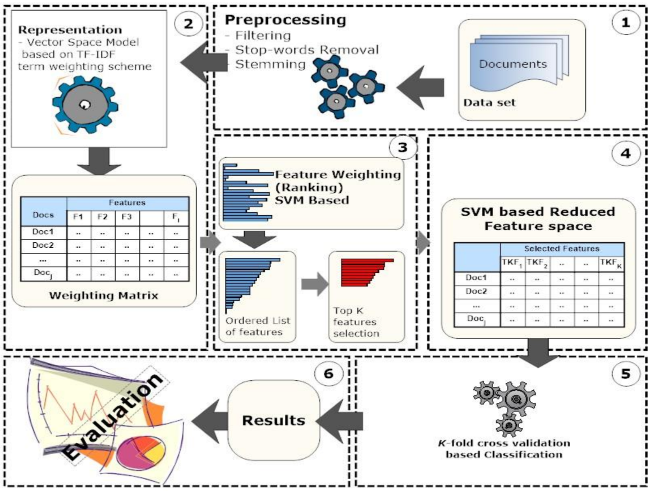

Fig. 1. Steps of text classification approach

C. *Step 3: SVM-FRM Application*

This step is the main contribution of this paper, where the SVM learning algorithm is utilized to rank the features of the feature space. The SVM algorithm assigns a weight value to each feature as a result of the training process. In normal usage of SVM as a classifier, these weights help the SVM to learn the hyper-plane that separates positive examples from negative examples of the dataset. However, in this paper, these weights are employed as a ranking method of the features. The features ranked high are assumed to be more distinguished and so lead to better classification results. The output of applying SVM for feature ranking will produce a weight assigned to each feature in the unordered list. Thus, the list of features will be ordered in descending order according to the value of the weight. Then, Top K features will be selected for the next step.

D. *Step 4: Reduced Feature Space Construction*

After selecting the Top K features, the reduced feature space, based on these features, should be established before the classification. The construction of the reduced feature space is performed using the algorithm shown in Fig. 2.

E. *Step 5 – Classification*

In this step, the constructed reduced feature space is passed to the classification algorithm. Usually, the feature space can be treated into two ways. One is splitting it into two parts known as Training and Testing parts. The training part is used to train the classifier to construct the classification model, while the testing part is used to measure the performance of the constructed model.

```
Input:
        Feature Space FS_mxn /* m is the count of rows,
        and n is the count ofcolumns.*/
        List of Top K features.
Output:
        Reduced Feature Space RFS_mxk, /* m is the count
        of rows, and k is the count of selected features
        (columns).*/
Start
        For each column in FS
           FI = get Feature Identifier
           If FI exists in (List of Top K features)
              Do
                 Append column to RFS
              End Do
           End If
        Loop
End
```

Fig. 2. Reduced feature space construction algorithm

The other way is to split the matrix into K equal (or almost equal) parts known as folds (usually 10 folds). Then, the classification training and testing processes are performed in K rounds. In each round, one fold is considered as Testing, while the remaining *K-1* folds are used for Training. In this case, the performance is calculated by calculating the average of the performances obtained from all rounds. This research follows the second way and applies the classification based on the stratified 10-folds cross validation model [35] using the Support Vector Machine (SVM) classifier.

F. *Step 6: Evaluation*

Commonly, in TC domain, the performance metrics such as Precision, Recall, F-measure, and Accuracy are used to measure the "exactness", " completeness" and "correctness" respectively of the approach. Thus, they are quite helpful in providing an overall evaluation of the performance of the presented classification approach. However, literature review shows that high precision and recall values are hard to be achieved simultaneously as low values of recall may be the price of obtaining high levels of precision and vice versa [2]. This research considered the Accuracy metric in addition to averaged F-measure metric as the weighted harmonic mean of precision and recall for evaluation. Generally, text classification or categorization is a multiple class classification problem, in which the Precision, Recall, and F-measure metrics are calculated per class using the formulas 4 – 7.

$$P_{ci} = |TP_{ci}| / [|TP_{ci}| + |FP_{ci}|] \quad (4)$$

$$R_{ci} = |TP_{ci}| / [|TP_{ci}| + |FN_{ci}|] \quad (5)$$

$$F\text{-}measure_{ci} = 2 * [(P_{ci} * R_{ci}) / (P_{ci} + R_{ci})] \quad (6)$$

$$Accuracy_{ci}=(|TP_{ci}|+|TN_{ci}|)/(|TP_{ci}|+|FP_{ci}|+|FN_{ci}|+|TN_{ci}|) \quad (7)$$

Where P_{ci}, R_{ci} are the Precision and Recall of class *ci*, respectively. TP_{ci} is the count of documents correctly labeled to be in class *ci*, and FP_{ci} is the number of documents incorrectly labeled by the classifier to be in class *ci*. FN_{ci} is the number of documents incorrectly identified not to be in class *ci*, and TN_{ci} is the number of documents correctly labeled not to be in class *ci*. In spite of the fact that text classification is usually considered as a multi-class classification problem, the averaged F-measure is calculated in this research based on the formula (8), where *n* is the number of classes (i.e categories) in the dataset.

$$\text{Averaged F-measure} = 2 \times \left[\frac{\left[\frac{\sum_{i=1}^{n} P_{ci}}{n}\right] \times \left[\frac{\sum_{i=1}^{n} R_{ci}}{n}\right]}{\left[\frac{\sum_{i=1}^{n} P_{ci}}{n}\right] + \left[\frac{\sum_{i=1}^{n} R_{ci}}{n}\right]} \right] \quad (8)$$

IV. Datasets and Experiments

A. *Datasets*

In order to evaluate the performance of the proposed method, experiments were conducted on three common Arabic text classification collections: BBC, Watan, and Abuiadah datasets. These datasets were selected to test the proposed method in different situations such as balance and dataset size in terms of the number of documents. A brief description of these corpora is provided next. Fig. 3 shows the statistical distribution of documents in these datasets.

1) *Watan dataset [36]:*

This corpus contains more than 20000 documents that fall into six categories which are: culture, religion, economy, local news, international news and sports. Originally, the numbers of documents in these categories are not equal. Thus, the researchers select 9900 documents that are equally distributed over the categories. The aim of considering this dataset is testing the performance of the proposed method under the big sized dataset condition. In Arabic TC domain, this corpus is popular and have been used widely in many works such as in [37] and [38].

2) *Abuaiadah dataset [39]:*

This is a balanced dataset that consists of 2700 documents distributed equally in nine categories which are: economy, health, law, literature, politics, religion, sport, and technology. The documents of this corpus are of the same size (approximately 2 Kilobytes) and were collected from various resources. Even though this dataset is new, it has been used in many Arabic TC works such as [40], and [41].

3) *BBC dataset [42]:*

BBC is an unbalanced free dataset that consists of 4763 documents. The documents in this dataset are distributed in seven different classes which are: business and economy, middle east news, Misc, newspapers highlights, science and technology, sports, and world news. This dataset is used widely in Arabic TC such as in [43], and [44].

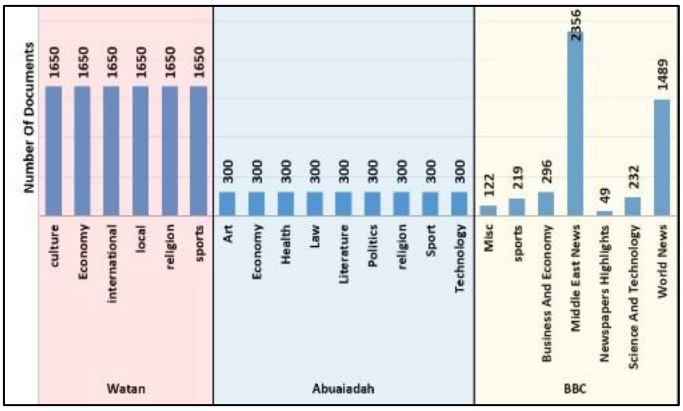

Fig. 3. Documents distribution in considered datasets

As seen in Fig. 3, the Watan and Abuaiadah datasets are balanced datasets (i.e. the count of documents are equal in all categories) with a different number of categories. It should be said that the Watan dataset is a big-sized dataset while Abuaiadah dataset is small-sized one. The BBC dataset, however, is an unbalanced dataset with an adequate number of documents.

B. *Experiments*

The proposed SVM based Feature Ranking Method (SVM-FRM) was tested against many of the commonly applied traditional feature ranking (selection) techniques such as Information Gain (IG), Correlation (Corr), and Chi Square (CHi2). Three datasets were used for the purpose of making the referred to comparison. The performance of SVM-FS was tested against these methods based on a different number of features. As explained in Step 3 on the proposed approach, all features in the feature space are ranked based on SVM-FRM and other ranking methods as well, individually. Feature subsets of different sizes were selected and considered for classification. The sizes of these feature subsets are 100, 500, and 1000 to 5000 features (in intervals of 500 features). The top *K* ranked features according to each of the ranking methods were selected each time and the experiment is carried out, the total number of classification processes were 144. The Rapidminer Studio V7.5 software was used to carry out these experiments. Fig. 4 shows the structure of the basic process in Rapidminer.

V. Results and Discusson

This section presents the accuracy and averaged F-measure results on the considered corpora.

Fig. 5, 6, and 7 show the accuracy results of experiments completed on the datasets Watan, Abuaiadah, and BBC, where the Full FS is the full feature space of each dataset which counts 86389, 43462, and 38630 features, respectively. TABLE I. shows the averaged F-measure benchmarking/comparison results.

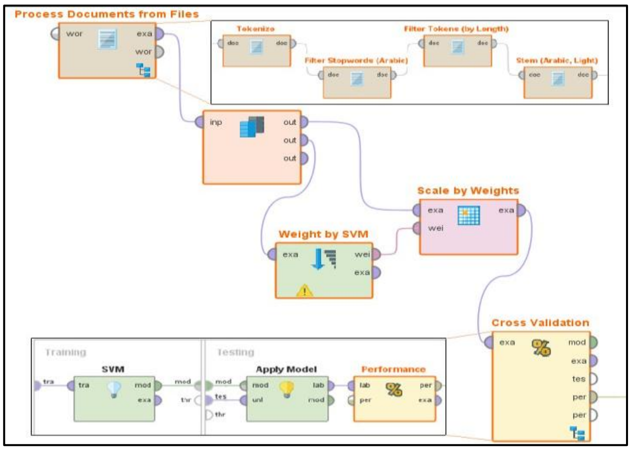

Fig. 4. Structure of SVM-FRM in Rapidminer

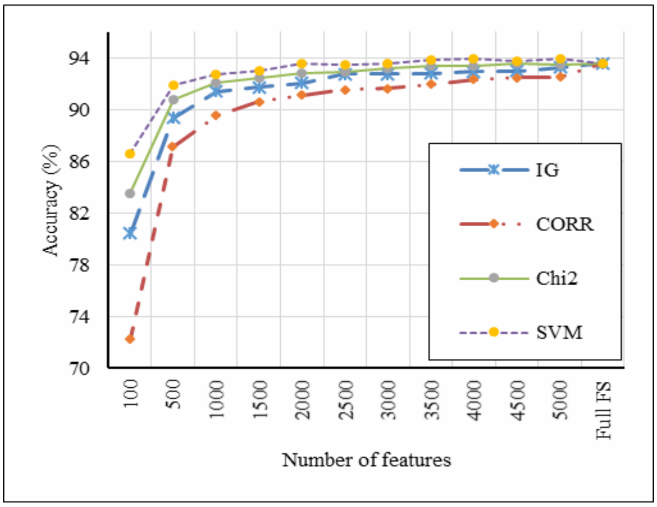

Fig. 5. Accuracy results on Watan dataset

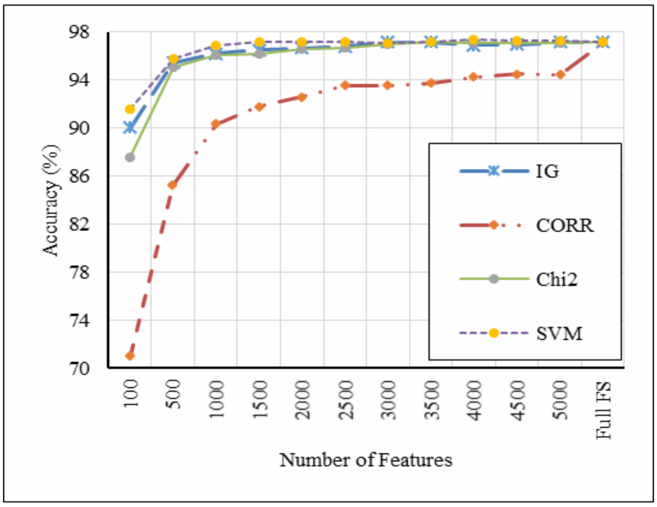

Fig. 6. Accuracy results on Abuaiadah dataset

TABLE I. AVERAGED F-MEASURE BENCHMARKING RESULTS

No. of Features	Watan				Abuaiadah				BBC			
	IG	*Cor*[a]	*Chi2*	*SVM*	*IG*	*Cor*[a]	*Chi2*	*SVM*	*IG*	*Cor*[a]	*Chi2*	*SVM*
100	81.27	73.32	83.85	86.76	90.46	72.66	88.65	91.79	76.01	52.87	66.87	49.52
500	89.43	87.23	90.84	91.91	95.49	86.09	95.09	95.83	84.59	65.12	79.81	66.31
1000	91.41	89.55	92.11	92.70	96.26	90.95	96.11	96.87	85.63	67.11	84.12	68.64
1500	91.73	90.64	92.45	93.02	96.53	92.08	96.25	97.16	86.10	67.51	85.06	68.98
2000	92.09	91.17	92.79	93.57	96.68	92.83	96.58	97.21	87.31	75.30	85.73	78.98
2500	92.77	91.56	92.93	93.57	96.83	93.73	96.73	97.24	87.04	76.89	87.12	84.25
3000	92.74	91.64	93.19	93.83	97.19	93.73	96.98	97.12	86.92	77.45	**87.50**	84.15
3500	92.83	91.97	93.40	93.90	97.16	93.91	97.16	97.23	87.56	80.78	87.07	84.73
4000	92.95	92.33	93.35	93.90	96.93	94.43	97.08	**97.38**	87.68	80.78	86.84	86.10
4500	92.98	92.47	93.53	93.75	96.97	94.66	97.05	97.27	87.17	81.83	86.14	86.04
5000	93.25	92.54	93.47	**93.94**	97.16	94.62	97.08	97.30	87.05	81.62	86.65	86.57
Full FS	93.61	93.61	93.61	93.61	97.20	97.20	97.20	97.20	86.52	86.92	66.87	86.92

[a] Correlation.

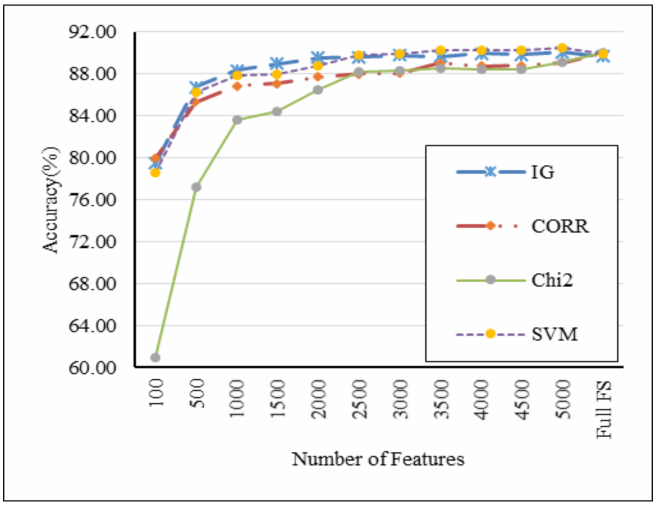

Fig. 7. Accuracy results on BBC dataset

As seen in Fig. 5 and Fig. 6, the proposed SVM-FRM outperform other traditional feature ranking methods on Watan and Abuaiadah datasets, with a maximum accuracy of 93.94% and 97.37% respectively. Documents in each of these datasets are distributed equally over dataset's categories (i.e. balanced datasets). However, the Watan dataset can be considered as big sized dataset as it consists of 9900 documents with a full feature space of 86389 features. The other dataset (i.e. Abuaiadah dataset), however, is a small dataset that contains less than 40000 features. The results in Fig. 5 and Fig. 6 indicate the ability of the proposed method to perform well in the condition of balanced datasets in spite of the dataset's size.

Results in Fig. 7 (i.e. accuracy results on BBC dataset) show that the proposed method outperforms the Correlation and Chi square feature ranking methods only. Similar to Abuaiadah dataset, BBC dataset is a small dataset with less than 40000 features. Still, BBC is an unbalanced dataset where the counts of documents in dataset's categories are not equal. In this case, our experimental results show that the IG feature ranking method outperforms other methods for the subsets that consist of less than 2500 features, while the proposed SVM-FRM outperforms other methods for the subsets that contain 3500 features and more with a maximum accuracy value of 90.49%.

Besides, the accuracy results based on the Full FS (i.e. full feature space) are equal in spite of the feature ranking method per dataset. This case indicates that all features in the feature space are included in the classification process, where a large number of noisy and less important features are considered, leading to a very long learning and classification time.

The reported averaged F-measure benchmarking results in TABLE I show that the proposed SVM-FRM not only outperforms other methods on the Watan and Abuaiadah datasets but also obtains superior f-measure performance, with maximum average F-measure values of 93.94% and 97.38%, respectively. On the contrary, none of the benchmarked methods shows superior average F-measure performance on the unbalanced BBC dataset, the maximum average F-measure value is obtained by the Chi2 method based on the feature set of size 3000 features.

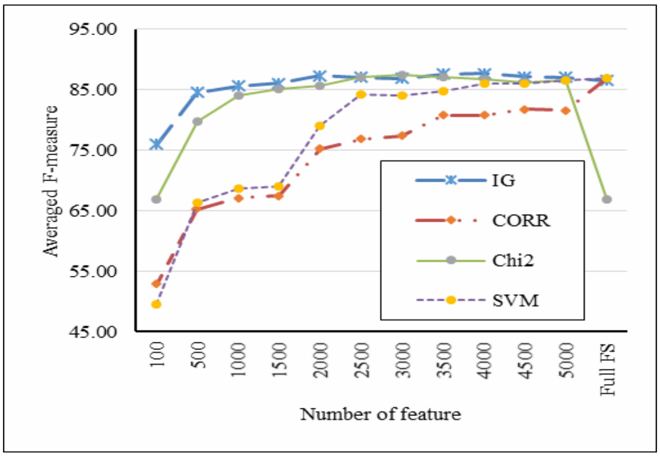

Fig. 8. Averaged F-measure benchmarking results on BBC dataset

However, the performance of SVM-FRM shows close results against the IG and Chi2 methods based on big-sized feature sets (i.e. feature sets contained of 5000 features and more), as shown in Fig. 8.

The presented accuracy and average F-measure results can lead to a conclusion that the proposed SVM-FRM has an outstanding performance in the case of balanced datasets (such as Watan and Abuaiadah datasets), while it shows comparatively less performance when applied on the unbalanced dataset.

VI. Conclusions and Future Work

In this paper, the researchers presented the Support Vector Machine based Feature Ranking Method (SVM-FRM), in which the weighting and ranking of features are based on the SVM learning algorithm. The benchmarking accuracy and average F-measure results with many different statistical feature selection methods on various datasets can lead to a conclusion that the proposed SVM-FRM has an outstanding performance in the case of balanced datasets (such as Watan and Abuaiadah datasets), while it shows less performance on unbalanced datasets. The future work of the research team will focus on improving the proposed method, so that it can perform higher on unbalanced datasets, along with examining its performance with different classification algorithms.

Acknowledgment

The authors would like to extend their thanks to Ms. Fatina Aljamal for her contribution to this work through improving and proofreading the manuscript.

References

[1] S. Jiang, G. Pang, M. Wu, and L. Kuang, “An improved K-nearest-neighbor algorithm for text categorization,” Expert Systems with Applications, vol. 39, pp. 1503-1509, Jan 2012.

[2] L. Man, C. L. Tan, S. Jian, and L. Yue, “Supervised and Traditional Term Weighting Methods for Automatic Text Categorization,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, pp. 721-735, 2009.

[3] R. M. Alguliyev, R. M. Aliguliyev, and N. R. Isazade, “An unsupervised approach to generating generic summaries of documents,” Applied Soft Computing, vol. 34, pp. 236-250, 2015.

[4] A. Tejeda-Lorente, C. Porcel, J. Bernabé-Moreno, and E. Herrera-Viedma, “REFORE: A recommender system for researchers based on bibliometrics,” Applied Soft Computing, vol. 30, pp. 778-791, 2015.

[5] M. Melucci, “Vector-Space Model,” in Encyclopedia of Database Systems, L. Liu and M. T. ÖZsu, Eds., ed Boston, MA: Springer US, 2009, pp. 3259-3263.

[6] T. Sabbah, A. Selamat, M. H. Selamat, R. Ibrahim, and H. Fujita, “Hybridized term-weighting method for Dark Web classification,” Neurocomputing, vol. 173, Part 3, pp. 1908-1926, 2016.

[7] V. Sulic, J. Perš, M. Kristan, and S. Kovacic, “Efficient dimensionality reduction using random projection,” in Proceedings of the Computer Vision Winter Workshop, Nove Hrady, Czech Republic, 2010, pp. 29-36.

[8] M. Efron, J. Zhang, and G. Marchionini, “Comparing feature selection criteria for term clustering applications,” in Proceedings of ACM SIGIR 2003, Toronto, Canada, 2003, pp. 28-31.

[9] K. K. Bharti and P. K. Singh, “Hybrid dimension reduction by integrating feature selection with feature extraction method for text clustering,” Expert Systems with Applications, vol. 42, pp. 3105-3114, 2015.

[10] G. Salton and C. Buckley, “Term-weighting approaches in automatic text retrieval,” Information Processing & Management, vol. 24, pp. 513-523, 1988.

[11] H. Wu, L. Robert Wing Pong, K. Wong, and K. Kwok, “Interpreting TF-IDF term weights as making relevance decisions,” ACM Transactions on Information Systems, vol. 26, pp. 1-37, 2008.

[12] M. Lan, C.-L. Tan, and H.-B. Low, “Proposing a new term weighting scheme for text categorization,” in Proceedings of the 21st national conference on Artificial intelligence, Boston, MS, USA, 2006, pp. 763-768.

[13] L. Luying, K. Jianchu, Y. Jing, and W. Zhongliang, “A comparative study on unsupervised feature selection methods for text clustering,” in Natural Language Processing and Knowledge Engineering, 2005. IEEE NLP-KE '05. Proceedings of 2005 IEEE International Conference on, 2005, pp. 597-601.

[14] L. Yanjun, L. Congnan, and S. M. Chung, “Text Clustering with Feature Selection by Using Statistical Data,” IEEE Transactions on Knowledge and Data Engineering, vol. 20, pp. 641-652, 2008.

[15] A. Selamat and S. Omatu, “Web page feature selection and classification using neural networks,” Information Sciences, vol. 158, pp. 69-88, Jan 2004.

[16] S. Robertson, “Understanding inverse document frequency: on theoretical arguments for IDF,” Journal of Documentation, vol. 60, pp. 503-520, 2004.

[17] W. Shang, H. Huang, H. Zhu, Y. Lin, Y. Qu, and Z. Wang, “A novel feature selection algorithm for text categorization,” Expert Systems with Applications, vol. 33, pp. 1-5, 2007.

[18] S. S. R. Mengle and N. Goharian, “Ambiguity measure feature-selection algorithm,” Journal of the American Society for Information Science and Technology, vol. 60, pp. 1037-1050, 2009.

[19] A. Onan, “Classifier and feature set ensembles for web page classification,” Journal of Information Science, vol. 42, pp. 150-165, 2016.

[20] Y. Jung, H. Park, and D. Du, “A Balanced term-weighting scheme for improved document comparison and classification,” preprint, 2001.

[21] M. Sanderson and I. Ruthven, “Report on the Glasgow IR group (glair4) submission,” in Proceedings of the The Fifth Text Retrieval Conference (TREC-5), Gaithersburg, Maryland, 1996, pp. 517-520.

[22] F. Crestani, M. Sanderson, M. Theophylactou, and M. Lalmas, “Short queries, natural language and spoken document retrieval: Experiments at Glasgow University. In: E.M. Voorhees and D.K. Harman (eds), The Sixth Text REtrieval Conference (TREC-6), 667–86. [NIST Special Publication 500–240] Available at: http://trec.nist.gov/pubs/trec6/papers/glasgow.ps.gz (accessed 5 December 2005).”, ed, 1998.

[23] H. Peng, L. Fulmi, and C. Ding, “Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, pp. 1226-1238, 2005.

[24] Y. Yang, “Noise reduction in a statistical approach to text categorization,” in Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval, Seattle, Washington, USA, 1995, pp. 256-263.

[25] T. Sabbah, A. Selamat, M. H. Selamat, F. S. Al-Anzi, E. H. Viedma, O. Krejcar, *et al.*, “Modified frequency-based term weighting schemes for text classification,” Applied Soft Computing, vol. 58, pp. 193-206, 2017.

[26] C. Lee and G. G. Lee, “Information gain and divergence-based feature selection for machine learning-based text categorization,” Information Processing & Management, vol. 42, pp. 155-165, 2006.

[27] Y. Yang and J. O. Pedersen, “A Comparative Study on Feature Selection in Text Categorization,” in Proceedings of the Fourteenth International Conference on Machine Learning (ICML'97), Nashville, TN, USA, 1997, pp. 412-420.

[28] A. K. Uysal, “An improved global feature selection scheme for text classification,” Expert Systems with Applications, vol. 43, pp. 82-92, 2016.

[29] G. Forman, “An extensive empirical study of feature selection metrics for text classification,” Journal of machine learning research, vol. 3, pp. 1289-1305, 2003.

[30] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," Journal of machine learning research, vol. 3, pp. 1157-1182, 2003.

[31] Y.-T. Chen and M. C. Chen, "Using chi-square statistics to measure similarities for text categorization," Expert Systems with Applications, vol. 38, pp. 3085-3090, 2011.

[32] D. Meyer and F. T. Wien, "Support vector machines," The Interface to libsvm in package e1071, 2015.

[33] L. Lee, C. Wan, R. Rajkumar, and D. Isa, "An enhanced Support Vector Machine classification framework by using Euclidean distance function for text document categorization," Applied Intelligence, vol. 37, pp. 80-99, 2012.

[34] Y. Aphinyanaphongs, L. D. Fu, Z. Li, E. R. Peskin, E. Efstathiadis, C. F. Aliferis, *et al.*, "A comprehensive empirical comparison of modern supervised classification and feature selection methods for text categorization," Journal of the Association for Information Science and Technology, vol. 65, pp. 1964-1987, 2014.

[35] W. Zhang, T. Yoshida, and X. Tang, "A comparative study of TF*IDF, LSI and multi-words for text classification," Expert Systems with Applications, vol. 38, pp. 2758-2765, 2011.

[36] M. Abbas, K. Smaili, and D. Berkani, "Comparing TR-Classifier and KNN by using Reduced Sizes of Vocabularies," Culture, vol. 1, p. 210, 2009.

[37] M. Abbas, K. Smaïli, and D. Berkani, "Evaluation of Topic Identification Methods on Arabic Corpora," JDIM, vol. 9, pp. 185-192, 2011.

[38] A. H. Aliwy, "Tokenization as Preprocessing for Arabic Tagging System," International Journal of Information and Education Technology, vol. 2, p. 348, 2012.

[39] D. Abuaiadah, J. El Sana, and W. Abusalah, "On the impact of dataset characteristics on arabic document classification," International Journal of Computer Applications, vol. 101, 2014.

[40] D. Abuaiadah, "Using Bisect K-Means Clustering Technique in the Analysis of Arabic Documents," ACM Trans. Asian Low-Resour. Lang. Inf. Process., vol. 15, pp. 1-13, 2016.

[41] I. Hmeidi, M. Al-Ayyoub, N. A. Abdulla, A. A. Almodawar, R. Abooraig, and N. A. Mahyoub, "Automatic Arabic text categorization: A comprehensive comparative study," Journal of Information Science, vol. 41, pp. 114-124, 2015.

[42] M. K. Saad and W. Ashour, "OSAC: Open Source Arabic Corpora," in Proceedings of the 6th International Conference on Electrical and Computer Systems, Lefke, Cyprus, 2010, pp. 118-123.

[43] S. A. YOUSIF, V. W. SAMAWI, I. ELKABANI, and R. ZANTOUT, "Enhancement Of Arabic Text Classification Using Semantic Relations With Part Of Speech Tagger," W transactions Advances In Electrical And Computer Engineering, pp. 195-201, 2015.

[44] G. Raho, G. Kanaan, and R. Al-Shalabi, "Different Classification Algorithms Based on Arabic Text Classification: Feature Selection Comparative Study," International Journal of Advanced Computer Science and Applications Ijacsa, vol. 6, pp. 23-28, 2015.