Automatic Kurdish Text Classification Using KDC 4007 Dataset

Tarik A. Rashid[1(✉)], Arazo M. Mustafa[2], and Ari M. Saeed[3]

[1] Department of Computer Science and Engineering,
University of Kurdistan Hewler, Erbil, Kurdistan, Iraq
tarik.ahmed@ukh.edu.krd

[2] School of Computer Science, College of Science,
University of Sulaimania, Sulaymaniyah, Kurdistan, Iraq
arazo.2007@yahoo.com

[3] Department of Computer Science, College of Science,
University of Halabja, Halabja, Kurdistan, Iraq
arimohsaeed@gmail.com

Abstract. Due to the large volume of text documents uploaded on the Internet daily. The quantity of Kurdish documents which can be obtained via the web increases drastically with each passing day. Considering news appearances, specifically, documents identified with categories, for example, health, politics, and sport appear to be in the wrong category or archives might be positioned in a nonspecific category called others. This paper is concerned with text classification of Kurdish text documents to placing articles or an email into its right class per their contents. Even though there are considerable numbers of studies directed on text classification in other languages, and the quantity of studies conducted in Kurdish is extremely restricted because of the absence of openness, and convenience of datasets. In this paper, a new dataset named KDC-4007 that can be widely used in the studies of text classification about Kurdish news and articles is created. KDC-4007 dataset its file formats are compatible with well-known text mining tools. Comparisons of three best-known algorithms (such as Support Vector Machine (SVM), Naïve Bays (NB) and Decision Tree (DT) classifiers) for text classification and TF × IDF feature weighting method are evaluated on KDC-4007. The paper also studies the effects of utilizing Kurdish stemmer on the effectiveness of these classifiers. The experimental results indicate that the good accuracy value 91.03% is provided by the SVM classifier, especially when the stemming and TF × IDF feature weighting are involved in the preprocessing phase. KDC-4007 datasets are available publicly and the outcome of this study can be further used in future as a baseline for evaluations with other classifiers by other researchers.

1 Introduction

In recent years, there is an enormous amount of machine readable data stockpiled in files and databases in a form of text documents. Text classification is one of the most common and convenient techniques for information exchange, in the meanwhile, much of the world's data can be found in text forms such as newspaper articles, emails,

L. Barolli et al. (eds.), *Advances in Internetworking, Data & Web Technologies*,
Lecture Notes on Data Engineering and Communications Technologies 6,
DOI 10.1007/978-3-319-59463-7_19

literature, web pages, etc. The rapid growth of the text databases is due to the increased amounts of information available in electronic forms such as e-mails, the World Wide Web, electronic publications, and digital libraries. Text mining can be defined as the process of discovering meaningful and interesting linguistic patterns from a large collection of textual data, and it is relevant to both information retrieval and knowledge discovery in databases [1–3].

In general, data mining is an automatic process of finding useful and informative patterns among large amounts of data or detecting new information in terms of patterns or rules from that enormous amount of data. Data mining usually deals with structured data, but information stored in text files is usually unstructured and difficult to deal with, and to deal with such data, a pre-processing is required to convert textual data into an appropriate format for automatic text processing. The purpose of text mining is to process unstructured textual, extract non-trivial patterns or meaningful pattern from the text, make the information included in the text accessible to the different data mining algorithms and reduce the effort required from users to obtain useful information from large computerized text data sources [1, 3]. Text mining generally multidisciplinary domain, thus, research works in texts involve dealing with problems such as text representation, text analysis, text summarization, information retrieval, information extraction, text classification and document clustering. In all these problems, data mining and statistical techniques are used to process textual data [1, 2].

One of the most widely utilized procedures in the text mining studies is the procedure of text classification which is addressed in general as is the task of learning under the supervisor. Text classification is a process of automatically classifying unstructured documents into one or more pre-defined categories such as science, art or sport… etc., based on linguistic features and content. The procedure can be depicted as a natural language problem and the aim of this paper is to reduce the need of manually organizing the huge amount of text documents. In the field of text classification problem, very few research works have been studied for the Kurdish Sorani language. Therefore, this field is at initial stages. It should be noted that due to the progress of the World Wide Web, and the increased number of non-English users, many research efforts for applying pre-processing approaches for other languages have been documented in literature. One of the most criteria in this framework is applying the text classification problem on Kurdish Sorani text documents.

2 Literature Survey

Since Kurdish Sorani script is considered as the closest to the Arabic language, and technically both have a written system that is from right to left. Thus, in this literature study, the research works in the text classification field have been sorted starting with the Kurdish language, shadowed by Arabic language, and then followed by English language.

Mohammed et al. in 2012 used the N-gram Frequency Statistics for classifying Kurdish text. An algorithm called Dice’s measure of similarity was employed to classify the documents. A corpus of Kurdish text documents was build using Kurdish Sorani news which consisted of 4094 text files divided into 4 categories: art, economy, politics, and sport. Each category was divided equally per their sizes (50% as a training set and 50% as a testing set). The Recall, Precision and F1-measure were used to compare the performance. The results showed that N-gram level 5 outperformed the other N-gram levels [4].

In [5], Al-Kabi M et al., in 2011, conducted comparison between three classifiers Naïve Bayes classifier, Decision Tree using C4.5 Algorithm and Support Vector Machine to classify Arabic texts. An in-house collected Arabic dataset from different trusted websites is used to estimate the performance of those classifiers. The dataset consisted of 1100 text documents and divided into nine categories: Agriculture, Art, Economics, Health, Medicine, Law, Politics, Religion, Science, and Sports. Additionally, pre-processing (which included word stemming and stop words removing) was conducted. The experiments showed that three classifiers achieved the highest accuracy in cases that did not include stemming. While the accuracy was decreased when using stemming. This means that the stemming had impacted negatively on the performance of the classification accuracy of the three classifiers.

In [6], Mohammad AH et al., in 2016 studied the performance of three well-known machine learning algorithms Support vector machine, Naïve Bayes and Neural Network (NN) on classifying Arabic texts. The datasets consisted of 1400 Arabic documents divided into eight categories collected from three Arabic news articles namely: Aljazeera news, Saudi Press Agency (SPA), Alhayat. In terms of performance, three evaluation measures were used (recall, precision and F1-measure). The results indicated that SVM algorithm outperformed NB and NN and F1-measure for three classifiers were 0.778, 0.754, and 0.717 respectively.

In [7], Mohsen AM et al., in 2016 conducted study to compare the performance of different well known machine classifiers to classifying emotion documents. The ISEAR dataset was applied. It consisted of 7,666 documents belonging to five categories namely: Anger, Disgust, Fear, Joy and Guilt. Tokenization, stop word removal, stemming and lemmatization as preprocessing tasks and TF-IDF as term weighting. Also, two lexicons were used which are NRC emotion lexicons (National Research Council of Canada) and SentiWordNet sentiment lexicons. Based on the obtained results, the authors concluded that Logistic Model Tree (LMT) is the most appropriate classifier in comparison with the other algorithms for English emotion documents classification.

3 Text Mining Functionalities

In this paper, three types of classification techniques are used as described in the following subsections.

3.1 Naive Bayes Classifier

Naive Bayes classifier is a probabilistic based approach which is based on Baye's theorem. It is a simple and efficient to implement [8, 9]. Naive Bayes classifier underlies on the assumption that the features (words) in the dataset are conditionally independent which is computing the probability of each by figuring the frequency of features (words) and the relevance between them in the dataset [8]. Despite that the features independence assumption is unrealistic, Naive Bayes has been discovered extremely effective for many functional applications, for example, text classification and medical diagnosis, even when the dimensionality of the input is high [9]. Advantages of NB would include simplicity, efficiency, robustness and interpretability, while the main disadvantage of NB is that it does not work properly with data having noises. Thus, remove all the noises before applying NB classifier is the need [10].

3.2 Decision Tree Classifier

Decision tree algorithm is widely used in machine learning and data mining. It is also simple and can be easily understandable and converted into a set of humanly readable if-then rules [11]. The decision tree mechanism is used to test some feature values of unseen instances at each node for classifying or finding the class of a given unseen instance where the test starts at the root node and goes down to a leaf node. Information gain is a suitable measure for choosing the best feature where the feature with highest information gain is chosen to be the root node [12]. Advantages of the decision tree can include straightforwardness, interpretability and capacity to handle feature interactions. In addition, the decision tree is nonparametric, which makes issues like exceptions and whether the dataset is linearly divisible [8]. Disadvantages of the decision tree can include the lack of support for online learning and suffer from the issue of over fitting, which can be handled using different strategies like random forests (or boosted trees) or perhaps the problem of over fitting could be avoided by pruning the tree [8].

3.3 Support Vector Machine Classifier

SVM is a supervised machine learning algorithm and was proposed for text classification by [13]. Researchers have used SVM widely in a text categorization task, such as in [8]. In N-dimensional space, input points are mapped into a higher dimensional space and then a maximal separating hyperplane is found. SVM technique classification depends on the Structural Risk Minimization principal [14]. The linear Kernel function is used in this paper scope as there is a very large number of features in the document classification problem. Thus, SVM is suitable for text categorization problems due to their ability to learn [8]. Advantages of SVM can involve high accuracy, and has great theoretic guarantees about overfitting [8]. Similarly, they work well regardless of whether the data is linearly separable or not. Disadvantages of SVM can involve complexity, poor interpretability and high memory requirements.

4 Methods and Materials

Before plunging into the details of the used methods and material in this paper, it is worth mentioning an overview about Kurdish language. The Kurdish language belongs to the Indo-European family of languages. This language is spoken in the geographical area spanning the intersections of Iran, Iraq, Turkey, and Syria [15]. However, Kurds have lived in other countries such as Armenia, Lebanon, Egypt, and some other countries since several hundred years ago, [16]. The Kurdish language is generally divided into two widely spoken and most dominant dialects, namely; Sorani and Kurmanji. Kurdish is written using four different scripts, which are modified Persian/Arabic, Latin, Yekgirtû (unified), and Cyrillic [16]. The Persian/Arabic script is mainly used in Sorani dialect for writing. Kurdish Sorani text documents have used in this research. The Sorani text is more complex with its reading from right-to-left and its concatenated writing style. The Kurdish Sorani character set is consisted of 33 letters which are shown in Fig. 1.

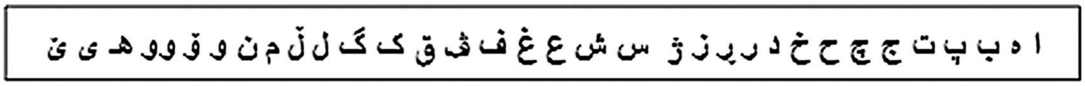
ا ه ب پ ت ج چ ح خ د ر ڕ ز ژ س ش ع غ ف ڤ ق ک گ ل ڵ م ن و ۆ وو هـ ی ێ

Fig. 1. The Kurdish Sorani alphabets.

In the next subsections, Kurdish Sorani pre-processing steps, data representation and term weighting are explained.

4.1 Kurdish Sorani Pre-processing Steps

Dataset pre-processing is an important stage in text mining. A huge number of features or keywords in the documents can lead to a poor performance in terms of both accuracy and time. For the problem of text classification, a document, which typically has high dimensionality of feature space and most of the features (i.e., terms) are irrelevant to the classification task or non-informative terms. The main objective of the pre-processing steps is to prepare text documents which are represented by many features for the next step in text classification. The proposed model of the pre-processing steps for Kurdish Sorani text documents was introduced by authors in [17]. The most common steps for the Kurdish pre-processing steps are tokenization, normalization, stop-word filtering, and Kurdish stemming. The proposed Kurdish stemming-step module of the pre-processing stage is a step-based approach to stages via which a word goes through before arriving at the extracted root of the word. This stemmer determines the words that have several affixes (e.g. a word that has 'prefix' + 'root' + 'suffix1' + 'suffix2' + ••• + 'suffixN'). This approach is not only utilized for stripping affixes from nouns and verbs as it is used in other languages, but it is also used to strip affixes from the stop words. A list that contains nearly 240 stop words (words that are widely used in Kurdish Sorani) has been used [17].

4.2 Data Representation and Term Weighting

The text representation model is a process of transforming a document from a series of characters into sequences of words so that to be appropriate for learning the algorithm and the classification task. The representation of text document can be coded as a form of a matrix, where columns indicate words that distinguish the objects (text documents) stored in each row, where each apparent word is a feature and the number of times the word occurs in the text document is its value [18]. The most common technique for text representation in the text classification task is the bag of words (BOW) [18]. This method of document representation is also known as Vector Space Model (VSM); this is the most common and easy way of text representation [19]. Each term that appears in documents must be represented to a machine learning classifier as real-number vectors of weights [19]. Thus, per a text classification research, the weighting of the term can be divided into three major approaches [20]:

4.2.1 Boolean or Binary Weighting

This is the simplest way of encoding for the term weighting. If the corresponding word (term) is used in the document d_j at least once, then it is set to 1 otherwise to 0 [19, 20].

4.2.2 Term Frequency (TF)

In term frequency weighting scheme, an integer value indicates the number of times that the term appears in a specific document d_j [19].

4.2.3 Term Frequency Inverse Document Frequency (TF × IDF)

TF-IDF can be considered as the most accurate application for text categorization domains with more than two categories. The TF-IDF weights are typically preferred over the other two options [20]. It is a straightforward and efficient method for weighting the terms in text documents categorization purposes [18]. In this work, the TF-IDF weighting function is used which is based on the distribution of the terms within the document, and within the collection, where the higher value indicates that the word occurs in the document, and does not occur in many other documents, and in inverse amount to the number of documents in the collection for which, the word occurs at least one time [20]. This is can be calculated as follows:

$$\mathrm{TF.IDF}\left(\mathrm{t_i},\mathrm{d_j}\right) = \mathrm{TF}\left(\mathrm{t_i},\mathrm{d_j}\right) * \log(\mathrm{N/DF(t_i)}) \tag{1}$$

where *TF* is the frequency of the term in document d_j and *DF* (t_i) is the number of documents that contain term t_i, after stopping word removal and word stemming, and N is the total number of documents.

5 Dataset, Experimentations, and Evaluation

Details of data set, experimental studies, different test options and various evaluation metrics are explained in the following subsections.

5.1 Dataset

Since datasets in general are not accessible for tests and text classification studies. Thus, this new dataset called KDC-4007 is created. The most important feature of this dataset is its simplicity and it is well-documented. The data set can be accessed and widely used in various studies of text classification regarding Kurdish Sorani news and articles. The documents consisted of eight categories, which are Sports, Religions, Arts, Economics, Educations, Socials, Styles, and Health, each of which is consisted of 500 text documents, where the total size of the corpus is 4,007 text files. The dataset and documents can become freely accessible to have original outcomes via future experimental assessments on KDC-4007 dataset (KDC-4007 dataset can be accessed through: https://github.com/arazom/KDC-4007-Dataset/blob/master/Kurdish-Info.txt). Table 1, gives a full detail about the KDC-4007 dataset version. In datasets, the ST-Ds is just stop words elimination is performed by using Kurdish preprocessing-step approach. In the Pre-Ds dataset, Kurdish preprocessing-step approach is used. In the Pre + TW-Ds dataset, $TF \times IDF$ term weighting on Pre-Ds dataset is performed. In the Orig-Ds datasets, no process is used which is original dataset.

Table 1. KDC-4007 Dataset Experimentation.

Dataset Name	K-Preprocessing-Step Module	Stop-word Filtering	TF-IDF Weighting	No. of DOC's	# of Features
Orig-Ds	No	No	No	4,007	24,817
ST-Ds	No	Yes	No	4,007	20,150
Pre-Ds	Yes	Yes	No	4,007	13,128
Pre + TW-Ds	Yes	Yes	Yes	4,007	13,128

5.2 Experimentations

Generally, the point of performing text classification is to classify uncategorized documents into predefined categories. However, when we look from machine learning point of view, the objective of text classification is to learn classifiers from labeled documents and satisfy categories on unlabeled documents. In literature, there is an affluent set of machine learning classifiers for text classification. The determination of the best performing classifier relies on various parameters, for example, dimensionality of the feature space, number of training examples, over-fitting, feature independence, straightforwardness and system's requirements. Taking into consideration the high dimensionality and over-fitting aspects, three well-known classifiers (C4.5, NB and SVM) are chosen among all classifiers in our experimentation.

Each classifier is tested with the 10-fold cross validation technique, which is a very common strategy for the estimate of classifier performance, the data is divided into 10 folds; nine folds of the data are used for the training, and one-fold of the data is used for testing.

5.3 Evaluation

In the field of machine learning, there are diverse evaluation criteria that can be used to appraise classifiers. In this study, the four popular evaluations; accuracy (ACC), precision, recall and F1-measure are utilized. Their mathematical equations are illustrated below:

$$\mathrm{Accuracy} = \mathrm{TN} + \mathrm{TP}/\mathrm{TP} + \mathrm{FP} + \mathrm{TN} + \mathrm{FN} \tag{2}$$

$$\mathrm{Precision} = \mathrm{TP}/\mathrm{TP} + \mathrm{FP} \tag{3}$$

$$\mathrm{Recall} = \mathrm{TP}/\mathrm{TP} + \mathrm{FN} \tag{4}$$

$$\mathrm{F} - \mathrm{Measure} = 2 * (\mathrm{Recall} * \mathrm{Precision})/(\mathrm{Recall} + \mathrm{Precision}) \tag{5}$$

Accuracy it is the most widely used on a large scale to assess the standard of performance, which is the proportion of the total number of class files that are properly classified. In addition, the time to build the model is involved in the comparatives analysis. The classifiers compare the effectiveness of the proposed approach to measure how accurate the classification was by counting the number of correctly classified instances and the number of incorrectly classified instances. It is worth noticing that the same datasets are applied on all classifiers.

6 Results and Discussion

In this section, three different classifiers are used to study the effect of each of the preprocessing tasks. The three classifiers are used with four different representations of the same datasets. After conducting comparisons on the datasets, some insightful thoughts and conclusions can be discussed. The objective of this set of experiments was to compare the performance of the considered classifiers for each of the four different tests of the dataset.

Tables 2, 3 and 4 show the accuracy for the four different representations with three classifiers, the number of correctly classified instances (CCI), the number of incorrectly classified instances (ICI), and time spent to build mode (TB). Per the proposed technique, using normalization, stop-word removal, and Kurdish Stemming-step module produced a positive impact on classification accuracy in general. As shown in Table 2, the Kurdish preprocessing-step module provided a dominant impact and generated a significant improvement (in terms of classification accuracy) with the SVM classifier. This can be seen from the experiences of the Pre-Ds and the Pre + TW-Ds datasets respectively. On the other hand, stop word removal provided a slight improvement with the SVM classifier which can be seen from the ST-Ds dataset. However, stemming helped in gathering the words that contained similar importance, a smaller number of features with further discrimination were achieved. For any classification system, the model building time is a critical factor. As expected, the learning (model building) times for the four tests were generally low compared with the NB and DT (C4.5). Utilizing Kurdish Stemming-step module reduced the building times for the classifier compared with the Orig-Ds dataset. In addition, the average precision and recall of the eight categories for the Pre-Ds were satisfactory compared to the Orig-Ds dataset in which stemming processing was used which reduced the size of feature that effected the final performance of Kurdish text classification.

Table 2. Accuracies for the SVM Classifier on KDC-4007 Dataset

Trails	ACC%	CCI	ICI	Precision	Recall	F1-Measure	TB (Sec)
Orig-Ds	87.17	3493	514	0.87	0.87	0.87	4:27
ST-Ds	87.62	3511	496	0.88	0.87	0.87	4:20
Pre-Ds	91.44	3664	343	0.92	0.91	0.91	3:33
Pre + TW-Ds	91.48	3666	341	0.92	0.91	0.91	4:23

On the other hand, it was noticed that the precision, recall and F-measure for the ST-Ds dataset were slightly effected. The Pre + TW-Ds results using the SVM with the TF-IDF term weighting yielded better than using the DT (C4.5) and the NB with the TF-IDF term weighting.

From Table 3, on the DT classifier, it can be concluded that the Pre-Ds had the best performance in general. On the other hand, the Pre + TW-Ds included feature-weighting $TF \times IDF$ and produced accuracy that was almost the same as the Pre-Ds. The results in the Orig-Ds (the original dataset is used) were very small, whereas, the performance for same dataset and the same classifier used with Pre-Ds dataset increased significantly compared to the Orig-Ds dataset. The reason for this is that the test in the preprocessing step contained Kurdish Stemming-step module technique; whereas, the performance for the ST-Ds increased marginally which contained stop word removal in the preprocessing stage. Another measure which obtained from the experiments was the amount of time taken for building the models. As shown in the Table 3, the DT required a huge amount of time to build the needed model for four different datasets in general. While the time for building the models in tests contained the preprocessing stage decreased very significantly compared to the original dataset. The weighted averages for the precision, recall and F1- measure in the Orig-Ds dataset are very small. Though the F1- measure for the same dataset and the same classifier used in the Pre-Ds and the Pre + TW-Ds increased significantly compared with the Orig-Ds dataset. The reason for this is that the two tests in preprocessing step contained stemming, thus it can be inferred that the Kurdish Stemming-step module improved the Precision and Recall for the classifier.

As indicated by the data introduced in Table 4 for the NB classifier, the highest accuracy (86.42%) achieved when the pre-processing steps were used with the Pre-Ds dataset. After performing feature weighting $TF \times IDF$, the NB classifier obtained the worst accuracy results with the Pre + TW-Ds compared to the values obtained from the Pre-Ds dataset, where they were unexpected. As expected, the building times for classifier like the NB, required a small amount of time to complete the model compared to the DT. Consequently, it can be noticed from Table 4, that the results gave the highest average Precision, Recall and F1- measure when the pre-processing steps were used.

Table 3. Accuracies for the DT Classifier on KDC-4007 Dataset

Trails	ACC%	CCI	ICI	Precision	Recall	F1-Measure	TB (Sec)
Orig-Ds	64.88	2600	1407	0.65	0.64	0.65	231:33
ST-Ds	64.26	2575	1432	0.68	0.64	0.64	228:49
Pre-Ds	80.58	3229	778	0.81	0.80	0.80	150:29
Pre + TW-Ds	80.53	3227	780	0.81	0.80	0.80	164:29

Table 4. Accuracies for the NB Classifier on KDC-4007 Dataset

Trails	ACC%	CCI	ICI	Precision	Recall	F1-Measure	TB (Sec)
Orig-Ds	76.89	3081	926	0.77	0.76	0.77	9:36
ST-Ds	79.13	3129	881	0.78	0.78	0.78	14:5
Pre-Ds	86.42	3464	544	0.86	0.86	0.86	10:36
Pre + TW-Ds	82.48	3305	702	0.82	0.82	0.82	10:50

Also, it was noticed that when the feature-weighting *TF* × *IDF* was used, the F1-measure was decreased for the same dataset and the same classifier.

The dataset was experimented using 10-fold cross validation method. As illustrated in Fig. 2, the best result obtained was through the SVM classifier. From the experimental results, as in Fig. 2, it is obvious that the Kurdish preprocessing-step module technique significantly influenced the performance of the DT classifier on the four datasets. Thus, the range of accuracy in the DT was higher than SVM classifier.

In other words, the dimension of the dataset less influences the range of accuracy in the SVM classifier than the DT and NB classifiers. This is because it works better in a high dimensional environment.

Figure 3, shows the performance comparison of feature weighting *TF* × *IDF* methods in terms of accuracy on datasets. The accuracy performance values of the two classifiers excluding the SVM on datasets were insignificantly decreased after applying feature weighting *TF* × *IDF* method. The accuracy values in the Pre-Ds for the NB

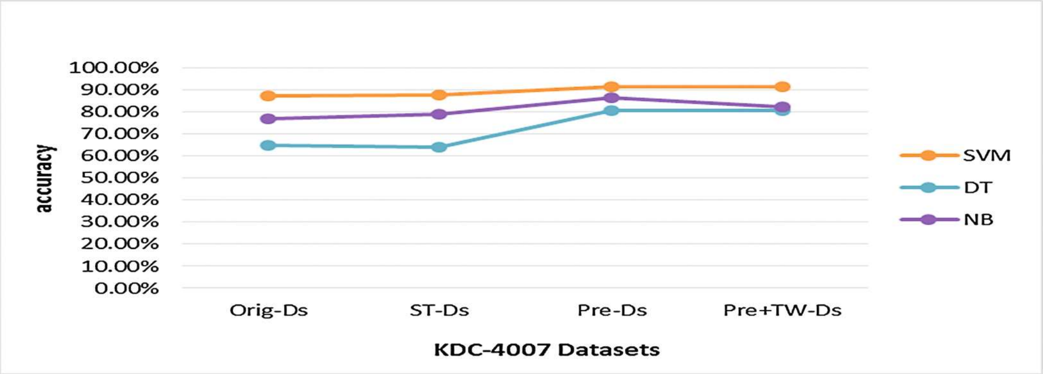

Fig. 2. SVM, NB and DT Results on the four versions of the dataset using Fold = 10.

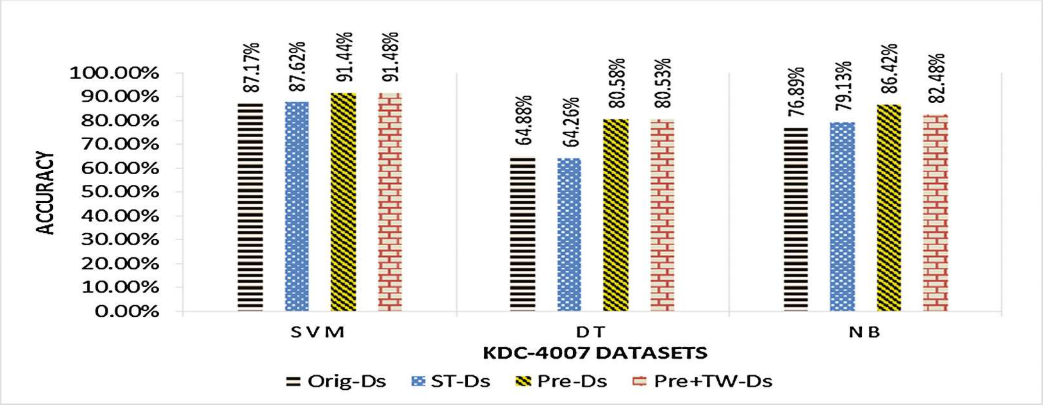

Fig. 3. Feature Weighting TF × IDF on Datasets using the SVM, NB and DT classifiers.

and DT classifiers were 86.42% and 80.58% respectively; however, they became 82.48%, and 80.53% after performing feature weighting TF × IDF method in the Pre + TW-Ds. The only classifier with insignificantly increased performance was the SVM classifier. For example, the accuracy value on the Pre-Ds datasets was increased from 91.44% to 91.48%.

7 Conclusion

In this research, the experiments indicated that the SVM outperformed both NB, and C4.5 classifiers in all tests. Applying normalization and Kurdish Stemming-step module on the original datasets was affected the performance of the three used classifiers, thus, these classifiers provided better classification accuracy compared to the original data. The performance of the classifiers SVM, NB and C4.5 was increased marginally when the stop word filtering approach was used in the preprocessing stage. Term weighting, such as *TF* × *IDF* method was performed after pre-processing steps to determine the impacts of feature weighting methods on Kurdish text classification. The experimental results indicated that; SVM increased the classification accuracy value by 0.25%, but, the classification accuracy was decreased by 5.1 using NB classifier. Besides, the classification accuracy was not affected when DT (C4.5) was used.

References

1. Hotho, A., Nurnberger, A., Paaß, G.: A brief survey of text mining. LDV Forum-GLDV J. Comput. Linguist. Lang. Technol. **20**, 19–62 (2005)
2. Tan, A.: Text mining: the state of the art and the challenges concept-based. In: Proceedings of the PAKDD 1999 Workshop on Knowledge Discovery from Advanced Databases, pp. 65–70 (1999)

3. Chen, K.C.: Text Mining e-complaints data from e-auction store. J. Bus. Econ. Res. **7**(5), 15–24 (2009)
4. Mohammed, F.S., Zakaria, L., Omar, N., Albared, M.Y.: Automatic kurdish sorani text categorization using N-gram based model. In: 2012 International Conference on Computer & Information Science (ICCIS), 12 Jun 2012, vol. 1, pp. 392–395. IEEE (2012)
5. Wahbeh, A., Al-Kabi, M., Al-Radaideh, Q., Al-Shawakfa, E., Alsmadi, I.: The effect of stemming on arabic text classification: an empirical study. Int. J. Inf. Retrieval Res. **1**(3), 54–70 (2011)
6. Mohammad, A.H., Alwada'n, T., Al-Momani, O.: Arabic text categorization using support vector machine, Naïve Bayes and neural network. GSTF J. Comput. (JoC) **5**(1), 108–115 (2016)
7. Mohsen, A.M., Hassan, H.A., Idrees, A.M.: Documents emotions classification model based on tf-idf weighting measure. World Acad. Sci. Eng. Technol. Int. J. Comput. Electric. Automat. Control Inf. Eng. **3**(1), 1795 (2016)
8. Hmeidi, I., Al-Ayyoub, M., Abdulla, N.A., Almodawar, A.A., Abooraig, R., Mahyoub, N. A.: Automatic Arabic text categorization: a comprehensive comparative study. J. Inf. Sci. **41** (1), 114–124 (2015)
9. Rish, I.: An empirical study of the naive Bayes classifier. In: IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence, 4 August 2001, vol. 3, no. 22, pp. 41–46. IBM, New York (2001)
10. Sharma, R., Gulati, N.: Improving the accuracy and reducing the redundancy in data mining. Int. J. Eng. Sci., 45–75 (2016)
11. Last, M., Markov, A., Kandel, A.: Multi-lingual detection of web terrorist content. In: Chen, H. (ed.) WISI. LNCS, pp. 16–30. Springer (2006)
12. Kotsiantis, S.B., Zaharakis, I., Pintelas, P.: Supervised machine learning: a review of classification techniques, vol. 31, pp. 249–268 (2007)
13. Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learn. **20**(3), 273–297 (1995)
14. Burges, C.J.: A tutorial on support vector machines for pattern recognition. Data Min. Knowl. Discov. **2**(2), 121–167 (1998)
15. Esmaili, K.S., Eliassi, D., Salavati, S., Aliabadi, P., Mohammadi, A., Yosefi, S., Hakimi, S.: Building a test collection for Sorani Kurdish. In: Proceedings of the 10th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA 2013), Ifrane, Morocco, 27–30 May 2013. IEEE, New York (2013)
16. Hassani, H., Medjedovic, D.: Automatic kurdish dialects identification. Comput. Sci. Inf. Technol., 61 (2016)
17. Mustafa, A.M., Rashid, T.A.: Kurdish stemmer pre-processing steps for improving information retrieval. J. Inf. Sci., 1–14 (2017). doi: 10.1177/0165551510000000, sagepub.co.uk/journalsPermissions.nav, jis.sagepub.com
18. Szymański, J.: Comparative analysis of text representation methods using classification. Cybern. Syst. **45(**2), 180–199 (2014)
19. Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. Commun. ACM **18**(11), 613–620 (1975)
20. Patra, A., Singh, D.: A survey report on text classification with different term weighing methods and comparison between classification algorithms. Int. J. Comput. Appl. **75**(7) (2013)