International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 14 (2017) pp. 4309-4319
© Research India Publications. http://www.ripublication.com

Comparative Study of Five Text Classification Algorithms with their Improvements

Ahmed H. Aliwy[1] and Esraa H. Abdul Ameer[2]

[1]Faculty of Computer science and Mathematics, University of Kufa, Iraq.
E-mail: ahmedh.almajidy@uokufa.edu.iq, ahmed_7425@yahoo.com

[2]Faculty of Computer science and Mathematics, University of Kufa, Iraq.
E-mail: asraahussein66@gmail.com

Abstract

Text classification is one of the important fields in natural language processing. It has many applications in the commercial world like email spam filtering, information retrieval and many other applications. There are many algorithms were used in text classification where few of them are essential. Decision Tree, Support Vector Machine, K-Nearest Neighbors, Naïve Bayes and hidden Markov model are the most essential five classification algorithms. All these algorithms were modified, by many researchers, to obtain high precision. Our work is comprehensive study for almost all the amendments which were done on these five algorithms for text classification. Because of many researchers have been implement them on private data, the comparison for getting accurate decision is very difficult. These amendments are classified according to Learner (modification), main algorithm (modification and addition) and features (extraction and reduction). Then comparison among these modifications for each algorithm is done.

INTRODUCTION

Text mining (TM), nearly equal to text analytic, is the operation of extract high-quality information from text. This high-quality information can be derived by extracting of style and direction using some algorithms. Text mining usually involves three main processes: (i) structuring the input text by parsing, driving linguistic features, removing of others etc. (ii) Deriving patterns within the resulted structured data. (iii) Finally, evaluation and interpretation of the output. 'High quality' in text mining usually refers to some combination of relevance, novelty, and interestingness. Typical text mining tasks include text categorization, text clustering, concept/entity extraction, production of granular taxonomies, sentiment analysis, document summarization, and entity relation modeling (i.e., learning relations between named entities)" [1].

Text Classification is the most important process of TM and sub field from it. The classical text classification is working by assign correct class to new document from set of classes. Text classification model can predict one class ("single label") for each document or multiple classes ("multi labels") for each document according to the system requirements. TC technology combined with some information processing technologies such information filtration and search engine, which optimized the quality of information service effectively.

At present-day, the fundamentally common text classification methods are the Decision Tree (DT), Support Vector Machine (SVM), K Nearest Neighbors (KNN), Naïve Bayes (NB) and hidden Markov model (HMM). These five classification algorithm are recognized as a simple and effective methods of text classification. Our work focuses on these five classification algorithm where we will summarize them and comparing them according to their modifications and their own uses.

RELATED WORK

In the recent years, the progress of web and social network technologies have led to a massive interest in the classification of text documents containing links or other meta-information and many studies on classification algorithms have been done by many researches. In this section we will do a review to these works and show the focus points of them. As we will see, the novelty of our work is appears by studying almost all the modification and improvements to each algorithm.

Aggarwal & Zhai (2012) [2] Focused on specific changes which are applicable for the text classification. They used, as text classification algorithms, Decision Trees, Pattern (Rule)-based Classifiers, SVM Classifiers, Neural Network Classifiers, Bayesian (Generative) Classifiers, nearest neighbor classifiers, and genetic algorithm-based classifier. They discussed the methods for features selection in text classification and described these methods for text classification.

Korde & Mahender (2012) [3] Gave an introduction to text classification, process of text classification as well as the overview of the classifiers and tried to compare the some existing classifier on basis of few criteria like time complexity, principal and performance.

Colas & Brazdil (2006) [4] sought about old classification algorithms in text categorization. They, also, found systematically the weaknesses and strength of SVM, naive Bayes and KNN algorithms in text categorization and examined how the number of attributes of the feature space effected on the performance.

Mamoun & Ahmed (2014) [5] highlighted the algorithms that are applied to the text classification and gave a comparative study on different types of approaches to the text categorization. They compared between the accuracy of the utilized algorithms and its results.

Vala & Gandhi (2015) [6] described text classification process, compared various classifier and discussed feature selection method for solving problem of high dimensional data and application of text classification.

Pawar & Gawande (2012) [7] presented a comparative study on different types of approaches to text categorization. This study focused on understanding the systems that have demonstrated the best in terms of effectiveness alone with very large numbers of categories.

Bilski (2011) [8] described the most important techniques and methodologies used for the text classification. Effectiveness and advantages for contemporary algorithms are compared and their most applications presented. The used text classification algorithms are artificial neural networks, k Nearest Neighbor (kNN) approach, naive Bayes classifier, decision trees and rules induction algorithms.

Baharudin, Lee & Khan (2010) [9] provided a review of the theory and methods of document classification and text mining, focusing on the existing literatures.

Bhumika & Nayyar (2013) [10] highlighted the important algorithms that are employed in text documents classification, while at the same time making awareness of some of the interesting challenges that remain to be solved.

Gandhi & Prajapati (2012) [11] described and compared the three algorithms which are k-nearest neighbors classifier, naive Bayes and the Support Vector Machines. They defined the settings of the data which performed in experiments.

Our work described and compared the most essential five classification algorithms that are Decision Tree, Support Vector Machine, K-Nearest Neighbors, Naïve Bayes and hidden Markov model. We also surveyed the improvement which was done for each algorithm by many researchers. The improvements are divided into improvements on the types of the algorithm (learner, modification or/and addition) or features improvements (extraction or/and selection). These comparisons of the improvements of each algorithm give a novelty to our work.

Decision Tree

When decision tree is used for text classification it consist tree where internal node are label by term, branches represent weight and leaf represent the class. Tree can classify the document by running through the query structure from root until it reaches a certain leaf, which represents the goal for the classification of the document. "Most of training data will not fit in memory, decision tree construction it becomes inefficient due to swapping of training tuples"[12].

Decision trees (DT) are the widely utilized inductive learning methods. It is learned from labeled training documents. ID3 is one of the most well-known decision tree learning algorithms and it has extensions like C4.5 and C5. DT is a flowchart such as tree structures, each internal node indicate test on document, each branch acts outcome of the test, and each leaf node holds a class label. It has own advantages and drawbacks:

1. Advantages: Decision trees capable to learn disjunctive expressions and their robustness to noisy data seem convenient for document classification.
2. Disadvantages: learning of decision tree algorithms cannot guarantee to return the globally optimal decision tree [13].

Improvement

There are many improvements was done to DT algorithm itself, the learner and features also. These improvements can be modification / addition to the algorithm itself or extraction-selection/reduction of the features. 10 improvements, to DT, will be introduced in this section.

Vateekul & Kubat [14] worked on Imbalanced, Large Scale, and Multi-label Data where the computational is very difficult to implement using decision trees and it has costs. The researchers try to reduce these costs. They implemented FDT ("fast decision - tree induction"), uses a two parts technique: "(1) feature-set pre-selection and (2) induction of several trees, every for a different data subset".

Johnson, Oles, Zhang & Goetz (2002) [15] performed combination of a fast decision tree induction algorithm, suited to text data, and a modern method for converting a decision tree to a rule set which is simplified and logically equivalent to the original tree. Data sets used for comparing categorizers is the Reuters-21578 collection of categorized newswires which consist of a training set with 9603 items and a test set with 3299 items.

Lewis & Ringuette [16] Gated empirical results on the performance of decision tree learning and Bayesian classifier algorithm on two text categorization data sets. The first of them was a set of 21,450 Reuter's newswire stories. The second data set included of 1,500 documents of the U.S. Foreign Broadcast Information Service (FBIS) that used in the MUC-3 evaluation of natural language processing systems. Documents used a set of 8,876 binary features corresponding to English words occurring in 2 or more training documents. The features ranked for each category by using the information gain measure. The performance for this algorithm was reasonable where they showed that feature selection in the decision tree algorithm was particle effective in dealing with the large feature sets common in text categorization.

Harrag, El-Qawasmeh & Pichappan [17] used a decision tree algorithm which shows of classifying Arabic text documents. They suggested hybrid techniques of document frequency threshold by using embedded information gain criterion and the preferable feature selection criterion. They used two different corpora, the first Corpus was used a set of Arabic texts from different domains collected from the Arabian scientific encyclopedia. It contains 373 documents distributed over 8 categories. The Second Corpus was used a set of prophetic traditions collected from the Prophetic encyclopedia included 453 documents distributed over 14 categories. They got an accuracy of 0.93 for the scientific corpus and 0.91 for the literary corpus.

Badgujar & Sawant [18] utilized of L' Hospital Rule which eases the calculation process and improves the efficiency of decision making algorithm. They showed, in the result, the effect of improved C4.5 was better than the ID3 and C4.5 in three aspects such as node count, rule count and time complexity. The data was collected from UCI machine learning repository.

Galathiya, Ganatra & Bhensdadia [19] compared among ID3, C4. 5 and C5.0 then implemented the system. They showed that the efficiency of the new system was less complexity, high accurate, good speed, and low memory usage. They used cross validation, feature selection, reduced error pruning and model complexity along with the classification. The used dataset were Zoo dataset, Ionosphere, Contact-lenses, Au1_1000, Breast Cancer, iris, Annealing and Weather nominal dataset.

Galathiya, Ganatra & Bhensdadia [20] proposed C5.0 to implement the feature selection, cross validation, model complexity and reduce error pruning of the original C5.0 in order to reduce the error ratio. The reduced error pruning technique was used in the decision tree to solve over fitting problem. The classification error rate was reduced compared to the existing system and within less time. The decision tree is constructed by using RGUI with WEKA packages where the input to algorithm is a fixed set of attributes.

Pandya & Pandya [21] Compared ID3, C4.5 and C5.0 with each other. They found C5.0 gave among all these classifiers efficient result and it is more accurate. C5.0 utilized as the base classifier to classify with low memory usage and high accuracy. They used Cross-validation, in them test, which gave more reliable estimation. Relevant features selection and reduced error pruning technique were used with decision tree, where the accuracy of the system was gained 1 to 3%. The algorithm was implemented using WEKA packages.

Agrawal & Gupta [22] utilized of L' Hospital Rule that simplify the calculation process, improves the efficiency of a lot of decision making algorithm and improved the performance of existing algorithm in terms of time saving. The decision tree get speed up the growing, and also gated better information of rules using large amount of data collection. They used data from WEKA data mining tool.

Xu &Wang [23] based on Support vector domain description (SVDD) and problems of multi-class solved by improved SVM decision tree of text categorization. The SVM-DT was constructed to provide performance of superior multi-class classification. They showed that the performance and efficiency is good to this algorithm in classification precision. They used Reuters-21578 document collections which contains 21578 news articles as data set.

Support Vector Machine

The Support Vector Machine, which was proposed by Vapnik, provides "a maximal margin separating hyper plane" between two classes of data and has non-linear extensions[24]. It is a supervised classification algorithm which recently used successfully for many tasks of NLP as text classification [25][26].

SVM algorithm represents the text document as a vector where the dimension is the number of distinct keywords. If the document size is large then the dimensions are enormous of the hyperspace in text classification which causes high computational cost. The feature extraction and reduction can be used to reduce the dimensionality[27].

Improvement

There are many improvements and modifications done to SVM. These improvements increased the efficiency of SVM and hence the accuracy.

Ageev & Dobrov [28] analyzed the influence of different parameters for SVM on performance of text categorization and tuning the strategy for parameters depends on subject area. They used a sub-collection of RF legal from University Information System RUSSIA (include of 10372 documents). They used four parameters for improving SVM, Verification on Reuters-21578 dataset, Different kernel functions, Feature space reduction (about 80%) and the lastly was Relative weight of different errors. The performance was increased by

1-5% of the accuracy.

Amer, Goldstein & Abdennadher [29] applied two modifications: eta one-class SVMs and Robust one class SVMs to outliers to make one-class SVMs more suitable to unsupervised anomaly detection. The used datasets were obtained from the UCI machine learning repository, ionosphere, shuttle and satellite and the breast-cancer dataset.

Yao & Fan [30] Enhanced SVM style with a weighted kernel function depended on features of the training data for interference detection. Rough set theory was used to implement a feature standing and chosen task of the new style. The evaluation of the new style was done on the KDD dataset and the UNM dataset. They showed that “the suggested style perform better than the conventional SVM in accuracy, computation time, and false negative rate".

Rennie &Rifkin [31] compared the Support Vector and Naive Bayes Machines to the task of classifying multilayered text. They found that when using the support machine as portion of an ECOC scheme will very effects the task of classifying multilayered text. They used two well-known data sets, 20 Newsgroups and Industry Sector.

K-Nearest Neighbors

K-Nearest Neighbors (KNN) is known as simple and effective classifier of text categorization. The KNN classifier has three defects: the complexity of computing its sample similarity is huge; its performance is easily affected by single training sample and KNN doesn’t build the classification model since it is a lazy learning method. The complexity of KNN can be reduced by utilizing three ways, reducing dimension of vector text, reducing amount of training samples and fasting process of finding K nearest neighbors [32].

The KNN used to classifying document by calculating the distance between the document and all document in training set by using variation or similarity measure. Then finding the nearest K neighbors among all training documents and is assigned the document to the category which includes largest number of documents included in k nearest neighbors set [33].

Improvement

Many improvements were done to KNN algorithm. Some of these improvements will be explained in this section.

Yong ,Youwen & Shixiong [34] improved KNN text classification algorithm by: (i) Compressing the given training sets and deleting the samples near by the border, (ii) clustering the training sample sets of each category using k-means clustering algorithm, (iii) Introducing weight value which mention the significance of each training sample according of number of samples in the cluster which contains this cluster center, (iv) finally, using the modified samples to accomplish KNN text classification. They used training corpus of 19637 documents with 20 categories.

Barigou [35] used only a part of training set to classify a new instance satisfying the condition of relevance. This way worked on optimize classification precision and quicken classification time. This method was competitive in terms of predictive performance whereas the selection for minimum instance. The used experiment data obtained by testing two different data groups: the Reuters-21578 of 21578 documents and a 20-set newsgroup of 18288 documents.

Han, Liu, Shen & Miao [36] suggested an improvement to KNN way which noted as EKNN, to resolve "large-scale hierarchical classification problems". EKNN is two phase hierarchical text classification algorithm. Firstly KNN is applied to get top-k examples. Then several critical category-neighbors features are extracted and its weights are estimated. Finally, the categories prediction algorithm uses the optimal parameters to predict the categories for the testing documents. They used datasets like DMOZ, Wikipedia small, and Wikipedia Large dataset.

Peterson, Doom & Raymer [37] utilized KNN classifiers with varying similarity measures (cosine similarity, Euclidian distance and Pearson correlation) by using several datasets. They used four UCI datasets which represent real-world classification problems and were frequently utilized for compare newly developed algorithms.

Al-Shalabi & Obeidat [38] applied the KNN classifier with two tests. It was applied, in first test, with utilizing N-Gram (unigrams and bigrams) in the documents indexing. In the second test, they applied it with utilizing traditional single terms indexing method. The mean precision of utilizing N-grams and Single terms indexing were 73.57 and 66.88 respectively for the four Categories: Computer, Economics, Education and Engineer.

Naïve Bayes

The Naïve Bayes classifier is known as a group from simple probabilistic classifiers upon on a common supposition where all the features are freelance of each other, according to the category variable [39]. Naive Bayes was fast and easy for implemented, so was a base-line in text classification [40].

The Naïve Bayes is effective enough to classify the text in many domains, although it is less accurate than other discriminative methods as SVM [41].

Naive Bayes models the distribution of the documents in each class using a probabilistic model with independence assumptions about the distributions of different terms. It was a very prevalent method in the text classification area, where the binary independence classifier was one of the best known approaches to Naive Bayes classification which used binaryvalued vector representations of documents [42].

Improvement

Many improvements were done for NB classifier. Some of these improvements were modification of calculating of probability, feature reduction and little other characteristics. We will show some of these improvements in this section.

Singhal & Sharma [43] optimized the performance of Naive Bayes algorithms by removing the features that are redundant correlated before giving the dataset to classifier. This optimization is potential through the correlation based feature selection (CFS) algorithm as preprocessing to Naive Bayes classifier for training purpose. They proved their improvement on the selected dataset from the Tuned IT repository of machine learning databases.

Taheri, Mammadov & Bagirov [44] used conditional probabilities to finding dependency between features and apply it to Naïve Bayes classifier. They offered results of numerical experiments on 10 data sets obtained from UCI machine learning repository and LIBSVM. The results explained that the proposed algorithm significantly optimize the performance of the Naive Bayes classifier.

Petre [45] developed new version of Naive Bayes classifier without assuming independence of features. Edges were added between features that capture correlation among them which was important step in this algorithm. The proposed algorithm used conditional likelihood to finds dependencies between features. He used 10 data sets obtained from UCI machine learning repository and LIBSVM. In the results, the performance of the classifier has been improved and preserves its robustness, where this improvement becomes "more substantial when increased the size of the data sets".

Schneider [46] used simple transformation to simple modifications of the Naive Bayes text classifier. Simple transformation is working on removes duplicate words effectively in a document. Authoritative confidence scores were increased by viewing a training corpus as a clustering of the training documents and feature selection as a way to optimize that clustering. Four datasets were used: 20 Newsgroups, Web KB, Ling-Spam and Reuters-21578.

Kim, Rim, Yook & Lim [47] propositioned and evaluated some public and effectively techniques for optimizing the performance of naive Bayes text classifier. They suggested document style depending on parameter estimation and document length normalization. In addition, Mutual-Information-weighted naive Bayes text classifier is proposed to increase the effect of highly informative words. The used data set was Reuters21578 and 20 Newsgroups collections.

He & Ding [48] Used several smoothing ways included the absolute smoothing, linear smoothing, Good-Turing smoothing and Witten-Bell smoothing to estimation the parameters introduced during naive Bayes text classifier. They show that the suggested ways can be achieved best and more stable performance than Laplace smoothing. The used data set were "extracted 3,894,900 questions from Yahoo! Webscope dataset".

Yuan [49] optimized Naïve Bayes text classification by “calculating posterior probability and reducing dimension of feature words of text". The results for experiment specified that the enhanced way has higher efficiency than the original algorithm. The used data set was "the Starter Edition text classification data made by Sogou laboratory which has17910 documents of 9 categories".

Hidden Markov model

The Hidden Markov Model (HMM) knows a robust statistical tool of modeling obstetric sequences that recognized by an underlying process generating an observable sequence. HMM are used for many areas interested in signal processing, particular speech processing and many NLP tasks as phrase chunking, part-of-speech tagging, and extracting target information from documents [50].

It considered as a state diagram that include of a set of states and transformations between them. Each state work on an output observation with a certain probability for these HMMs was a “double random process”[51].

Improvement

HMM have many improvements especially in the calculating of probability and fasting the computation time using special algorithm as Viterbi algorithm. We will explain some of the improvements in HMM for text classification in this section.

Frasconi, Soda &Vullo [52] suggested more public formulations for text categorization, which letting to the documents be organized as sequences of pages. Where introduced a novel hybrid system more specifically prepared to multi-page text documents. This taking into account contextual information provided by the whole page sequence can help disambiguation and improves single page classification accuracy. They used two different data sets: a subset from the journal American Missionary and a subset of Scribner's Monthly.

Murugesan & Suguna [53] described the technique “Minimum Message length estimator”, for getting a most suitable Hidden Markov Model with optimize number from states. The MML estimator is introduced for optimizing the HMM for getting on highest probability. This model used in a biological sequence analysis problem.

Overall the improvements

As we showed in the previous sections, there are many improvements to the well-known algorithms for text classification. Summary of all improvements mentioned in this work is showed in table-1. The improvements are divided to two main classes: algorithm and features. The improvements in algorithm can be modification/addition to the algorithm and the learner. The improvements on the features can be in the extraction or/and reduction.

Algorithms improvements

Learner: Lewis & Ringuette[16], Harrag , El-Qawasmeh & Pichappan [17], Badgujar & Sawant[18], Galathiya, Ganatra & Bhensdadia [19], Galathiya, Ganatra & Bhensdadia [20], Pandya & Pandya [21], Agrawal & Gupta [22], Yao, Zhao & Fan [30], Rennie & Rifkin [31], Barigou [35], Han, Liu, Shen & Miao [36], Al-Shalabi & Obeidat [38], Maneesh Singhal & Rama Shankar Sharma[43], Yuan[49] improved the learner (only) of the algorithms.

Learner and the algorithm itself: Vateekul & Kubat [41], Xu & Wang [23], Ageev& Dobrov [28], Amer, Goldstein & Abdennadher [29], Yong, Youwen & Shixiong [34], Taheri, Mammadov & Bagirov [44], Petre [45] and Schneider [46] improved the learner and add improvements to the algorithm itself.

The algorithm itself (**only**): Johnson, Oles, Zhang & Goetz [15], Peterson, Doom & Raymer [37], Kim, Rim, Yook & Lim [47] , He & Ding [48], Frasconi, Soda & Vullo [52], Murugesan &Suguna [53] added improvements on the algorithm itself (only).

Improvements by Features

feature Extraction and reduction: Vateekul & Kubat [41] , Johnson, Oles, Zhang & Goetz [15] Lewis & Ringuette[16], Harrag, El-Qawasmeh & Pichappan [17], Agrawal & Gupta [22], Ageev& Dobrov[28] , Yao, Zhao & Fan [30], Rennie & Rifkin [31], Barigou [35], Han, Liu, Shen & Miao [36], Al-Shalabi & Obeidat [38], Peterson, Doom & Raymer [37], ManeeshSinghal & RamashankarSharma[43], Petre [45] and Yuan[49] added improvements to the feature in terms of Extraction and reduction.

Feature reduction: Badgujar & Sawant[18], Galathiya, Ganatra & Bhensdadia [19], Galathiya, Ganatra & Bhensdadia [20], Pandya & Pandya [21], Yong, Youwen& Shixiong [34], Taheri, Mammadov& Bagirov [44] and Frasconi, Soda & Vullo [52] added improvements to the features in terms of reduction (only) .

Feature extraction: Schneider [46] used the feature extraction and modified the used algorithm.

DISCUSSION

Text mining offers an interesting combination of text classification algorithms. From of them: Decision Tree, Support Vector Machine, K-Nearest Neighbors, Naïve Bayes and hidden Markov model are the most essential five classification algorithms. In this paper we have attempted to do a comparative study for these five text classification algorithms with almost all the amendments which were done on these algorithms. We have described each algorithm separately and studied the modifications made to the same algorithm. These improvements are classified according to Learner (modification), main algorithm (modification and addition) and features (extraction and reduction).

This study showed that the easiest way to improvement the classification is by using feature reduction which cause (i) fasting the classification beside of (ii) increasing the efficiency. Another reason which is modification of some algorithms is very difficult to reach.

Also, this study showed that modification of learner is straightforward and can help for increasing the accuracy of the algorithm.

From the table 1, we can see each researcher has own dataset for testing the improvement which make the comparison more difficult.

Table 1: The improvements to text classification algorithms.

algorithm	researcher	Improvement					Data Set
		algorithm			feature		
		learner	modification	addition	Extraction	reduction	
DT	Vateekul & Kubat [14].	✓	☒	✓	✓	✓	Different data subset.

	Johnson, Oles, Zhang & Goetz [15].	☒	☒	✓	✓	✓	The Reuters-21578 collection of categorized newswires
	Lewis & Ringuette [16].	✓	☒	☒	✓	✓	Two data sets: 1- Set of 21,450 Reuter's newswire stories. 2- 1,500 documents from the U.S. Foreign Broadcast Information Service (FBIS)
	Harrag, El-Qawasmeh & Pichappan [17].	✓	☒	☒	✓	✓	two different corpora; 1- Arabic texts from Arabian scientific encyclopedia of 373 documents from 8 categories. 2- Set of prophetic traditions or "Hadiths' collected from the Prophetic encyclopedia.
	Badgujar & Sawant [18].	✓	☒	☒	☒	✓	Data sets from UCI machine learning repository.
	Galathiya, Ganatra & Bhensdadia [19].	✓	☒	☒	☒	✓	Multiple dataset: Zoo dataset, Ionosphere, Contact-lenses, Au1_1000, Breast Cancer, iris, Annealing and Weather nominal dataset.
	Galathiya, Ganatra & Bhensdadia [20].	✓	☒	☒	☒	✓	Used RGUI with weka packages.
	Pandya & Pandya [21].	✓	☒	☒	☒	✓	Used weka packages.
	Agrawal & Gupta [22].	✓	☒	☒	✓	✓	Used large amount of data collection, data mining tool WEKA was used.
	Xu & Wang [23].	✓	☒	✓	☒	☒	Used Reuters-21578 collection.
SVM	Ageev& Dobrov [28].	✓	✓	☒	✓	✓	Collection as FRF-10372 consists of 10372 documents.
	Amer, Goldstein & Abdennadher [29].	✓	✓	☒	☒	☒	Datasets from the UCI machine learning repository, ionosphere, shuttle and satellite and the breast-cancer dataset.

	Yao, Zhao & Fan [30].	✓	☒	☒	✓	✓	The KDD dataset and the UNM dataset.
	Rennie & Rifkin [31].	✓	☒	☒	✓	✓	Use two well-known data sets, 20 Newsgroups and Industry Sector.
KNN	Yong, Youwen & Shixiong [34].	✓	✓	☒	☒	✓	Data from Chinese natural language processing group in Department of Computer Information and Technology in Fudan University of 19637 documents.
	Barigou [35].	✓	☒	☒	✓	✓	Used two different data sets: 1- Reuters-21578 data set that 2- 20 Newsgroups data set that
	Han, Liu, Shen & Miao [36].	✓	☒	☒	✓	✓	Used Dmoz, Wikipedia small, and Wikipedia Large dataset.
	Peterson, Doom & Raymer [37].	☒	✓	✓	✓	✓	Used biological or medical data with four UCI datasets
	Al-Shalabi & Obeidat [38].	✓	☒	☒	✓	✓	Private corpus collected from online Arabic newspapers archives including Al-Jazera, AlNahar, Al-Hayat, and Al-Dostor,
NB	ManeeshSinghal & RamashankarSharma [43].	✓	☒	☒	✓	✓	Used sample dataset ("Eucalyptus Soil Conservation ") from the TunedIT repository of "machine learning databases".
	Taheri, Mammadov& Bagirov [44].	✓	✓	☒	☒	✓	10 data sets from UCI machine learning repository and LIBSVM.
	Petre [45].	✓	☒	✓	✓	✓	Used the dataset concerns from UCI Machine Learning Repository.
	Schneider [46].	✓	✓	☒	✓	☒	Used four datasets: 20-Newsgroups, WebKB, Ling-Spam and Reuters-21578".
	Kim, Rim, Yook & Lim [47].	☒	✓	✓	☒	☒	Used the Reuters21578 and 20 Newsgroups collections.

	He & Ding [48].	☒	✓	✓	☒	☒	Extracted 3,894,900 questions from Yahoo! Webscope dataset.
	Yuan [49].	✓	☒	☒	✓	✓	17910 documents from the Starter Edition text classification data by Sogou laboratory.
Hmm	Frasconi, Soda & Vullo [52] .	☒	✓	✓	☒	✓	Used tow dataset, 1- A subset of the journal American Missionary. 2- A subset of Scribners Monthly.
	Murugesan &Suguna [53].	☒	✓	✓	☒	☒	Used "biological sequence analysis problem".

REFERENCES

[1] Text mining. (2017, May 5). In *Wikipedia, the Free Encyclopedia*. Retrieved 14:09, May 23,2017,from

[2] https://en.wikipedia.org/w/index.php?title=Text_mining&oldid=778865797.

[3] Aggarwal, C. C., & Zhai, C. (2012). A survey of text classification algorithms. In *mining text data* (pp. 163-222). Springer US.

[4] Korde, V., & Mahender, C. N. (2012). Text classification and classifiers: A survey. *International Journal of Artificial Intelligence & Applications*, *3*(2), 85.

[5] Colas, F., & Brazdil, P. (2006). Comparison of SVM and some older classification algorithms in text classification tasks. In *IFIP International Conference on Artificial Intelligence in Theory and Practice* (pp. 169-178). Springer US.

[6] Mamoun, R., & Ahmed, M. A. (2014). A Comparative Study on Different Types of Approaches to the Arabic text classification. In *Proceedings of the 1st International Conference of Recent Trends in Information and* (Vol. 2, No. 3).

[7] Vala, M., & Gandhi, J (2015). Survey of Text Classification Technique and Compare Classifier.

[8] Pawar, P. Y., & Gawande, S. H. (2012). A comparative study on different types of approaches to text categorization. *International Journal of Machine Learning and Computing*, *2*(4), 423.

[9] Bilski, A. (2011). A review of artificial intelligence algorithms in document classification. *International Journal of Electronics and Telecommunications*, *57*(3), 263-270.

[10] Khan, A., Baharudin, B., Lee, L. H., & Khan, K. (2010). A review of machine learning algorithms for text-documents classification. *Journal of advances in information technology*, *1*(1), 4-20.

[11] Bhumika, P. S. S. S., & Nayyar, P. A. (2013). A review paper on algorithms used for text classification. *International Journal of Application or Innovation in Engineering & Management*, *3*(2), 90-99.

[12] Gandhi, V. C., & Prajapati, J. A. (2012). Review on Comparison between Text Classification Algorithms. *International Journal*.

[13] Korde, V., & Mahender, C. N. (2012). Text classification and classifiers: A survey. *International Journal of Artificial Intelligence & Applications*, *3*(2), 85.

[14] Nalini, K., & Sheela, L. J. (2014). Survey on Text Classification. *International Journal of Innovative Research in Advanced Engineering*, *1*(6), 412-417.

[15] Vateekul, P., & Kubat, M. (2009, December). Fast induction of multiple decision trees in text categorization from large scale, imbalanced, and multi-label data. In *Data Mining Workshops, 2009. ICDMW'09. IEEE International Conference on* (pp. 320-325). IEEE.

[16] Johnson, D. E., Oles, F. J., Zhang, T., & Goetz, T. (2002). A decision-tree-based symbolic rule induction system for text categorization. *IBM Systems Journal*, *41*(3), 428-437.

[17] Lewis, D. D., & Ringuette, M. (1994, April). A

comparison of two learning algorithms for text categorization. In *Third annual symposium on document analysis and information retrieval* (Vol. 33, pp. 81-93).

[18] Harrag, F., El-Qawasmeh, E., & Pichappan, P. (2009, July). Improving Arabic text categorization using decision trees. In *Networked Digital Technologies, 2009. NDT'09. First International Conference on* (pp. 110-115). IEEE.

[19] Badgujar, M. G. V., & Sawant, K. (2016). Improved C4. 5 Decision Tree Classifier Algorithms for Analysis of Data Mining Application. *International Journal*, *1*(8).

[20] Galathiya, A. S., Ganatra, A. P., & Bhensdadia, C. K. (2012). Classification with an improved Decision Tree Algorithm. *International Journal of Computer Applications*, *46*(23), 1-6.

[21] Galathiya, A. S., Ganatra, A. P., & Bhensdadia, C. K. (2012). Improved Decision Tree Induction Algorithm with Feature Selection, Cross Validation, Model Complexity and Reduced Error Pruning. *International Journal of Computer Science and Information Technologies*, *3*(2), 3427-3431.

[22] Pandya, R., & Pandya, J. (2015). C5. 0 algorithms to improved decision tree with feature selection and reduced error pruning. *International Journal of Computer Applications*, *117*(16).

[23] Agrawal, G. L., & Gupta, H. (2013). Optimization of C4. 5 Decision Tree Algorithms for Data Mining Application. *International Journal of Emerging Technology and Advanced Engineering*, *3*(3), 341-345.

[24] Xu, Z., Li, P., & Wang, Y. (2012). Text classifier based on an improved SVM decision tree. *Physics Procedia*, *33*, 1986-1991.

[25] Rennie, J. D. (2001). *Improving multi-class text classification with naive Bayes* (Doctoral dissertation, Massachusetts Institute of Technology).

[26] Hotho, A., Nürnberger, A., & Paaß, G. (2005, May). A brief survey of text mining. In *Ldv Forum* (Vol. 20, No. 1, pp. 19-62).

[27] Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. *Machine learning: ECML-98*, 137-142.

[28] Guduru, N. (2006). *Text mining with support vector machines and non-negative matrix factorization algorithms* (Doctoral dissertation, University of Rhode Island).

[29] Ageev, M. S., & Dobrov, B. V. (2003). Support Vector Machine Parameter Optimization for Text Categorization Problems. In *ISTA* (pp. 165-176).

[30] Amer, M., Goldstein, M., & Abdennadher, S. (2013, August). Enhancing one-class support vector machines for unsupervised anomaly detection. In *Proceedings of the ACM SIGKDD Workshop on Outlier Detection and Description* (pp. 8-15). ACM.

[31] Yao, J., Zhao, S., & Fan, L. (2006, July). An enhanced support vector machine model for intrusion detection. In *International Conference on Rough Sets and Knowledge Technology* (pp. 538-543). Springer Berlin Heidelberg.

[32] Rennie, J. D., & Rifkin, R. (2001). Improving multiclass text classification with the support vector machine.

[33] Jiang, S., Pang, G., Wu, M., & Kuang, L. (2012). An improved K-nearest-neighbor algorithm for text categorization. *Expert Systems with Applications*, *39*(1), 1503-1509.

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

[35] Yong, Z., Youwen, L., & Shixiong, X. (2009). An improved KNN text classification algorithm based on clustering. *Journal of computers*, *4*(3), 230-237.

[36] Barigou, F. (2016). IMPROVING K-NEAREST NEIGHBOR EFFICIENCY FOR TEXT CATEGORIZATION. *Neural Network World*, *26*(1), 45.

[37] Han, X., Liu, J., Shen, Z., & Miao, C. (2011, September). An optimized k-nearest neighbor algorithm for large scale hierarchical text classification. In *Joint ECML/PKDD PASCAL Workshop on Large-Scale Hierarchical Classification* (pp. 2-12).

[38] Peterson, M. R., Doom, T. E., & Raymer, M. L. (2005, September). Ga-facilitated knn classifier optimization with varying similarity measures. In *Evolutionary Computation, 2005. The 2005 IEEE Congress on* (Vol. 3, pp. 2514-2521). IEEE.

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

[40] Xu, S. (2016). Bayesian Naïve Bayes classifiers to text classification. *Journal of Information Science*,

0165551516677946.

[41] Rennie, J. D., Shih, L., Teevan, J., & Karger, D. R. (2003, August). Tackling the poor assumptions of naive bayes text classifiers. In *ICML* (Vol. 3, pp. 616-623).

[42] Ting, S. L., Ip, W. H., & Tsang, A. H. (2011).Is Naive Bayes a good classifier for document classification. *International Journal of Software Engineering and Its Applications*, *5*(3), 37-46.

[43] Cachopo, A. M. D. J. C. (2007). *Improving methods for single-label text categorization* (Doctoral dissertation, Universidade Técnicade Lisboa).

[44] Maneesh Singhal#1, Ramashankar Sharma#2(2014). Optimization of Naïve Bayes Data Mining Classification Algorithm. International Journal for research in applied Science and Engineering Technology (I JRAS ET). Vol. 2 Issue VIII, August 2014.

[45] Taheri, S., Mammadov, M., & Bagirov, A. M. (2011, December). Improving naive Bayes classifier using conditional probabilities. In *Proceedings of the Ninth Australasian Data Mining Conference-Volume 121* (pp. 63-68). Australian Computer Society, Inc.

[46] Petre, R. (2015). Enhancing Forecasting Performance of Naive-Bayes Classifiers with Discretization Techniques. *Database Systems Journal*, *6*(2), 24-30.

[47] Schneider, K. M. (2005, February). Techniques for improving the performance of naive bayes for text classification. In *International Conference on Intelligent Text Processing and Computational Linguistics* (pp. 682-693). Springer Berlin Heidelberg.

[48] Kim, S. B., Rim, H. C., Yook, D., & Lim, H. S. (2002, August). Effective methods for improving naive bayes text classifiers. In *Pacific Rim International Conference on Artificial Intelligence* (pp. 414-423). Springer Berlin Heidelberg.

[49] He, F., & Ding, X. (2007, April). Improving naive bayes text classifier using smoothing methods. In *European Conference on Information Retrieval* (pp. 703-707). Springer Berlin Heidelberg.

[50] Yuan, L. (2010). An improved Naive Bayes text classification algorithm in Chinese information processing. *Science*, 267-269.

[51] Hidden Markov model. (2017, March 5). In *Wikipedia, the Free Encyclopedia*. Retrieved13:28, April2, 2017, from https://en.wikipedia.org/w/index.php?title=Hidden_Markov_model&oldid=768811108.

[52] Vieira, A. S., Iglesias, E. L., & Diz, L. B. Study and application of Hidden Markov Models in scientific text classification.

[53] Frasconi, P., Soda, G., & Vullo, A. (2002). Hidden Markov models for text categorization in multi-page documents. *Journal of Intelligent Information Systems*, *18*(2), 195-217.

[54] Optimization of Hidden Markov Model using Minimum Message Length Estimator. Murugesan N1, Suguna P2,by International Journal of Emerging Technology and Advanced Engineering Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 3, Issue 8, August 2013).