International Journal of Advanced Technology in Engineering and Science
Vol. No.4, Issue No. 01, January 2016
www.ijates.com
ijates
ISSN 2348 - 7550

RESEARCH ISSUES IN TEXT CATEGORIZATION BASED ON MACHINE LEARNING: A REVIEW

Sumanta Kashyapi[1], Dr. Madhu Kumari[2]

[1,2]Computer Science, NIT Hamirpur, (India)

ABSTRACT

A collection of textual data becomes useful only when the valuable information contained by it is extracted. Text mining is the process of efficiently obtaining the information from a large set of text document. Machine learning techniques provide tools to get high quality information from very large and sparse data. That is why machine learning algorithms are often deployed to solve various text mining problems. Textual data is mostly very sparse in nature and possesses a large number of attributes. So to analyze it efficiently the data must be grouped into smaller categories. This categorization problem of text mining is a challenging field of research in which many machine learning algorithms had been explored effectively. This paper presents a survey on text categorization based on machine learning methodologies and various issues pertaining it.

Keywords: Data Mining, Machine Learning, Text Categorization, Text Mining

I. INTRODUCTION

Textual data contains valuable information in form of documents. Unless the information is extracted from the textual form it is not much useful. Moreover when we want to develop any kind of automatic system in which information embedded in textual data is to be used then it is absolutely necessary to extract high quality information efficiently. Text mining extracts high quality information from textual data by using various techniques from machine learning, statistics, information retrieval systems etc. In fact text mining associates the unstructured text data and relational database which only accepts data in a structured form. But unlike statistical problems where abundance of data is helpful for obtaining accurate results, overwhelming abundance of text resources make the task of text mining rather difficult. Also automatic Natural Language Processing from which text mining draws the techniques to process text data has many limitations. Due to these reasons researchers have to face many common issues while working with text mining. That is why researches must be familiar with these issues and the existing methods to overcome them. In this paper we have tried to identify some of the most significant issues in a sub-category of text mining process that is text categorization. There are four steps involved in text mining process. Collection of text data, Analysis of the data, Interpretation and Information extraction. Text categorization belongs to the data analysis part of the process which organizes the collection of data and by giving labels to the documents it structures the unstructured text data. Text categorization in turn helps data analysis which is much easier to carry out when a well defined structured and organized document is to be processed. Moreover text mining being an interdisciplinary field draws methods and techniques from various fields of study. So it faces issues associated with each of those fields. But in this paper we emphasize

only on the issues faced while working with machine learning methods. [1] [2] [3] [4] [5]

II. RESEARCH ISSUES IN TEXT CATEGORIZATION USING MACHINE LEARNING

Researchers face some common issues while trying to develop a good text categorization scheme. Significant efforts had been made in order to tackle these issues.

2.1. High dimensionality of text data - The most common and effective approach for text mining large

$$A = [a_{ij}]R_{m\times n}$$

collection of text documents is the vector space model. In vector space model we represent the documents as a collection of vectors. For each vector, a word present in the collection of documents along with the measure of it's importance in a particular context constitutes a single component of the vector. According to [6] this representation is known as term document matrix and it is expressed in the following manner:

m = total no. of terms n = total no. of documents a_{ij} = importance of the term of term i in document j

As the words in the documents are considered feature of the vectors and as per observations it is typical to find tens of thousands of words in a moderate sized document, vectors of text mining problems has very large dimensions. Due to this large dimensionality of the vectors it is computationally expensive to compare these vectors with all it's features intact.

Many techniques are adopted to handle this issue of high dimensionality. Some techniques reduce the number of dimensions by removing the least significant features while other techniques convert or transform the set of features to a new and smaller set of features [7]. Effectiveness of either technique depends on the particular problem in question. Following are some of the works in which these techniques are examined.

In [6] centroids and least squares methods are used to reduce the dimensions of the text data space. They carried out 5 sets of experiments to compare the results using the full space of feature space with the results using the dimension reduction algorithms. Their results suggest the following:

1. Dimension reduction using the centroid and orthogonal centroid algorithm they have developed are computationally efficient than SVD based methods. 2. Too much reduction in dimensions causes significant loss of information. There exists an optimum number of reduced dimensions which must be estimated first.

In [8] k-means clustering algorithm is used to cluster high dimensional text data. They also explored a common issue faced while applying k-means algorithm and proposed a promising solution. The issue is that being a hill climbing approach, classical k-means algorithm often get stuck in a local optimum specially in case of small dataset. The algorithm proposed in the paper refines the result of k-means using local search before iteratively applying k-means again. Experiments carried out by them clearly shows improvement of k-means algorithm with the refinement as suggested in the paper.

In [9] random projection method to reduce dimensions is explored. It is an alternative way to other statistical dimension reduction reduction techniques and is very effective when it is not practical to use classic statistical methods due to high computational cost for large dataset.

All the methods discussed above are classified as feature reduction techniques. Apart from that feature transformation techniques are also used to reduce dimensions. Some of the popular feature transformation methods are Supervised LSI (Latent Semantic Indexing), Supervised clustering, Linear Discriminant Analysis

etc. [7]

2.2. Sparseness of text data - The words are denoted as feature in case of text data and very few words (except article, prepositions etc.) are repeated in the document or across different documents. So the feature vector constructed out of text data are generally found to be sparse. The sparsity of the text data is even greater when the corpora size is large. Typical sparsity of text data in a large collection of documents is found to be 95% - 99% [10]. This becomes a issue in terms of computational speed while comparing these sparse vectors with each other. We need specialized methods to deal with sparse data in order to maintain the efficiency.

In [11] sparse PCA technique, a variant of traditional PCA, is experimented. It is found to be very effective when text data is sparse enough so that number of feature is greater than number of samples. It also exploits the fact that in real-life data there is a exponential decrease of variance and for that many features can be eliminated during pre-processing. Also it is found that sparse PCA is computationally less expensive than it's traditional counterpart in contrary to earlier results.

2.3. Feature Selection - When it comes to selection of features we have many choices among various feature selection techniques. Every technique has some advantages and disadvantages and it depends on the specific text mining problem in hand and the empirical results obtained while choosing which technique to use. Following are some of the most common feature selection techniques:

Gini index is a very common feature selection tool which exploits a feature's discrimination level. Given a particular context if a feature (or word) is very specific to a certain subject then it can be said that the feature has

$$G(w)=\sum_{i=1}^{k} p_i(w)^2$$

a very high discriminative power. We measure this discriminative power with gini index. Formally gini index of a feature can be expressed as following manner:

where $p_i(w)$ is the conditional probability that the document is categorized under class i where word/feature w is present.

In [12] this feature selection tool is put into test with other three. As per the experiments carried out gini index based feature selection tool outperformed mutual information, information gain and chi-square method with improvement statistics of 28.5%, 19% and 9.2% respectively.

Entropy based methods are also popular for feature selection. It generally measures the amount of information a single feature is carrying. So the features having less entropy values can be eliminated because we gain little information by incorporating them. That is why it is also called information gain method. The information gain

$$I(w)=-\sum_{i=1}^{k} P_i log(P_i)+F(w)\sum_{i=1}^{k} p_i(w) log(p_i(w))+(1-F(w))\sum_{i=1}^{k}(1-p_i(w))log(1-p_i(w))$$

or entropy measure for a feature can be calculated using the following formula:

where P_i is the global probability of class i, F(w) is the fraction of documents containing word w and $p_i(w)$ stands for the same as previous

In [13] a comparative study is given of different feature selection algorithms including entropy method and a variance of entropy based model named as Entropy based Category Coverage Difference is put into use.

Unlike these two methods there are chi square statistic and mutual information approach which measures

correlation between the features (terms) and categories. Chi square method is calculated using the following formula:

$$\chi_{i^2}(w) = nF(w)^2(p_i(w) - P_i)^2 \ / (F(w)(1-F(w))P_i(1-P_i))$$

where n = total number of documents and $p_i(w)$, F(w) and P_i stands for the same as previous.
Whereas mutual information is calculated as:
Now a global estimation is made by taking the average or maximum value of chi square or mutual info as:

$$M_i(w) = log(p_i(w) / P_i)$$
$$\chi^2_{avg}(w) = \sum_{i=1}^{k} P_i \chi_{i^2}(w), \chi^2_{max}(w) = max_i \chi_{i^2}(w)$$

$$M_{avg}(w) = \sum_{i=1}^{k} P_i M_i(w), M_{max}(w) = max_i M_i(w)$$

and
As these measure shows the level of correlation between the word and category, we can get to know the significance of that particular word while assigning correct category to the document in question. In [14] extensive experiments were carried out with various feature selection methods. According to their results chi square method had the best performance.

2.4. Different Classifiers - While classifying the pre-processed documents we can find a wide range of classifier algorithms to choose from. This is a very profoundly studied field of text mining. Three of the most popular classifiers are discussed here.

Support Vector Machine - Support vector machines (SVM) come under supervised learning models that are applied to various classification tasks. Support vector machines represent an extension to nonlinear models of the generalized portrait algorithm [15]. Joachims paper on SVM is reviewed in this paper [16]. In that paper the author claimed the effectiveness of SVM over other classification techniques and presented a set of experiment which supports the claim. It has been found that in text classification most of the features are relevant. Also individual document vectors are sparse. Kivinen et al. [17] presented evidence that in case of problems having dense concepts but sparse instance, models like SVM are well suited. Most of the text classification task is linearly separable. SVM's goal is to find such linear separators. [18] Joachims described the experimental setup used to test his claim in his paper (Joachims 1998). The empirical evaluation is done on two test collections. The first one is the Reuters-21578 dataset (http://www.research.att.com/ lewis/reuters21578.html). The ModApte split is used to generate the corpus. The corpus consists of 9603 training documents and 3299 test documents. The correspondence between words and categories in this corpus is rather direct. The second test collection is taken from the Ohsumed corpus (ftp://medir.ohsu.edu /pub/ohsumed) . Here the correspondence between words and categories is less direct. From the 50216 documents the first 10000 are used for training and the second 10000 are used for testing. The classification task is slightly different from the previous corpus. In this case indexing with respect to a particular term is considered the classification task. Distinct terms are extracted out from the documents after stemming and stop-word removal. In the experiments carried out, four

other classification models are tried along with two variants of SVM. Precision/Recall-Break even Point is used as a measure of performance. These experiments conclude that SVMs consistently achieve better performance on categorization tasks than existing methods substantially and significantly. Joachims also concluded three major advantages of SVMs in classification tasks. These are: 1. Ability to generalize well in high dimensional feature spaces, so feature selection is not needed 2. It is robust, exhibits good performance in all experiments avoiding catastrophic failure like observed for the conventional methods on some tasks 3. SVMs possess automatic parameter settings, so parameter tuning is not needed

Artificial Neural Network - Apart from classic classification models Artificial Neural models also find it's application in the field of text categorization. Miguel E. Ruiz and Padmini Srinivasan wrote a paper on how ANN can be applied in text categorization and also described their experiments regarding the same. ANN is descended from the perceptron model. Perceptron model consists of one output node with a layer of input nodes. Also the connection from every input to the output node is weighted. The value of the weight is variable and during the training phase the model alters the weights iteratively to get the desired output. The output function of the perceptron in terms of the inputs : O = 1 if $\Sigma\, w_i I_i + \theta > 0$ and O = 0 otherwise.

The problem with these simple perceptron model was it is not able to solve non-linearly separable problems. The problem was a major setback for research on artificial neural network until Hecht-Nielson(1992) showed a network of artificial neurons having a hidden layer can learn any function. Ruiz and Srinivasan explored two broad categories of ANN in their paper [19]. They are Backpropagation network which is of supervised type and Counterpropagation network which is of unsupervised type. Backpropagation network - This supervised ANN is developed by Rumelhart, Hinton and Williams(1986). The model learns a function by several iterations. For each iteration it propagates the input through the network to get the output and then it adjusts corresponding weights to compensate the error in the output. Hence it is called backpropagation. Like the perceptron model, the activation function is:

$$N_j = \sum w_{ij} O_i + \theta_j$$

where O_i is the output vector for previous node of j. Based on the input value output is calculated as:

$$O_j = 1 / (1 + e^{-Nj})$$

While the network is going through it's training mode the weights are adjusted after the calculation of the output. The adjusted weight is calculated as:

$$w_{ijnew} = w_{ijold} + \beta (errdrv)_i O_j$$

Threshold associated with each unit is also adjusted as :

$$\theta_{ijnew} = \theta_{ijold} + \beta (errdrv)_j$$

Here is the learning rate and errdrv is the error derivative for the current node j. Errdrv is calculated as:

$$errdrv_j = O_j(1-O_j)(y_j - O_j)$$

$$errdrv_j = O_j(1-O_j)(\sum_k (errdrv)_k w_{ik})$$

Counterpropagation network - The counterpropagation network as developed by Hecht-Nielson consists of three layers in total. After the input layer there is a hidden layer named Kohonen layer which learns by unsupervised methods and there is a output layer named Grossberg layer. After the kohonen layer is stable (i.e. unsupervised learning phase is over) supervised learning methods are applied to adapt the grossberg layer. The kohonen layer of the network which is used in the experiments carried out by Ruiz and Srinivasan worked in winners take all fashion. During every iteration weights in the kohonen layer are adjusted as:

$$w_{new} = w_{old} + \alpha(x - w_{old})$$

where x is the input corresponding to the winning node and α is the learning rate which is decreased in training period. Lateral inhibition is performed in order to take into account only the winning node and it's neighbors. This is done by Mexican hat operation. The neighborhood size of the winning node goes on decreasing as the training progresses. When the neighborhood reaches 0 the training period for the kohonen layer completes. After the kohonen layer is established, training for grossberg layer initiates. This layer is trained in supervised mode. Corresponding to some input vector, output from the established kohonen layer is fed to the grossberg layer and it's output is stored. If the final output is exceeding the predetermined error margin, then weights are adjusted using the following formula.

$$V_{ijnew} = V_{ijold} + \beta(y_j - V_{ijold})k_i$$

where y_j is the desired output, is the training constant and ki is the output from kohonen layer. In the experiments carried out by Ruiz and Srinivasan both backpropagation and counterpropagation network were used and their results were compared. 2344 medline documents were used as the dataset for the experiment. Every document contained manually added MeSH terms based on which categorization was to be performed. Pre-processing was done by SMART system to extract the stem words and MeSH terms. As the range of frequencies of both the stem and MeSH terms were large, thresholding is done to cancel out too specific and too general terms. At last 1016 stem words were used for the input layer. For the counterpropagation network input, kohonen and grossberg layers consisted of 1016, 540 and 180 nodes respectively. During evaluation recall and precision was used. Following are the formulas using which these are calculated. Recall = a/(a + c) precision = a/(a + b) where

a = the system and the expert assigned the category

b = the system assigned the category, but the expert didn't

c = the system didn't assign the category, but the expert did

d = the system and the expert didn't assign the category

For b and c the error committed by the system is calculated as: error_rate = (b + c)/(a + b + c + d) Also F measure is calculated as:

$$F_\beta = (\beta^2 + 1)a \ / ((\beta^2 + 1)a + b + \beta^2 c)$$

Here F0 denotes precision and F is recall.

The result from the experiments clearly shows that when provided with enough examples, ANNs can be trained to perform text categorization tasks. Although the result shows advantage of the backpropagation network over the counterpropagation network, the knowledge acquired during the training phase of the counterpropagation can be translated to fuzzy rules, which is not possible in case of backpropagation. The authors left the testing of scalability of this approch as their future en devour.

Decision Tree - Decision tree is a decision support tool that makes use of a tree like data structure and helps to take correct decision based on events and their expected outcomes [20]. Decision trees can be utilized while implementing machine learning algorithms. Johnson et al. (2002) [21] published a paper on how they developed a system based on decision tree and symbolic rule for text categorization. Decision trees were being used for various machine learning problems including text categorization for a long time. But Johnson et al. took this approach one step ahead with the incorporation of symbolic rule set model which is derived from the initial decision tree. For text categorization task at first the decision tree is constructed using a fast decision tree construction algorithm. Fast decision tree construction algorithm is used in this case because it takes advantage of sparsity of the document feature vectors. Based on some predetermined condition decision tree is grown with every document feature vectors. After constructing the tree smoothing is performed. This is done to prune the large tree as it is observed that the smaller tree will probabilistically give much better result than the larger one. The tree weighting idea of data compression is used to perform this task. After the decision tree is constructed and smoothing is done it is converted in a set of symbolic rules which can be used directly for the making the decisions for categorization of the documents. A fully automated text categorization system module is developed using these methods which is named KitCat (tool kit for text categorization). Several experiments were carried on KitCat with Reuters-21578 collection of categorized newswires. The precision-recall measure (as described previously with ANN) is used to estimate the performance of the system. It is reported by the authors that for the mentioned data set micro-averaged precision of the system was found to be 87.0percent and micro-averaged recall was 80.5 percent, the average of these two values being 83.8 percent. The system took 80 seconds for the training phase to complete on a 400 megahertz Pentium II processor. They also used KitCat generated human readable symbolic rules to study and categorize corporate websites. It is observed that the KitCat system worked very well on this Web site structure, with a prediction accuracy of about 86 percent. It is also applied to categorize email documents in which the performance measures found to be 92.8 percent as micro-averaged precision and a micro-averaged recall of 89.1 percent. Later these findings were used in IBM's web sphere products and eventually evolved into the IBM Text Analyzer product offering.

Other than the classifiers discussed here there are Probabilistic and Naive Bayes classifiers (Multivariate Bernoulli Model, Multinomial Model etc.), Regression-Based Classifiers, Proximity-based Classifiers and many more. Like the feature selection issue comparative study [16],[21] of the different classifiers based on empirical results must be carried out before choosing one of them. [22]

2.5. Stemming and stop word removal - Generally documents are pre-processed to some extent before using it as a input for classifiers. Among several pre-processing tasks stemming is the most important. Stemming stands

for the task of converting every word into it's root or stem. So 'computer', 'computing' and 'computation' are converted to their root word that is 'comput'. This is done because it is observed that in most of the cases different parts of speech of same word or different tense does not effect it's context for which the word is used. So by removing these extra pieces we shrink the size of the input by considerable amount. Porter's algorithm is used very effectively for stemming. In [23] the working of the algorithm is explained and an improvement to the existing algorithm is also proposed. They have also showed with empirical results how pre-processing in general positively affects text mining. Apart from their proposal of improving porter's algorithm many other attempts were made to achieve the same thing and this remains an open problem in this area. Other than porter's algorithm other algorithms are often used for stemming. In [24] a comparative study on various stemming algorithm can be found. Like stemming, stop word removal is also done to reduce the input size. Stop words such as articles, prepositions, conjunctions etc. are connectives which are used to connect words or to emphasize the meaning. Proper grammatical rules must be followed while using stop words. But stop words do not contribute much in text mining tasks. So these stop words are found out from the text and removed to make the input text precise. To find the stop words from a text, every word in the text is matched with a list of stop words. It can be argued that by stemming and removing the stop words the meaning of the text is altered. In most of the cases a pre-processed text sentence may not carry any meaning at all. In some cases it results in ambiguities due to which the subsequent text mining module makes wrong interpretation. So recently semantic preserving pre-processing techniques are given more preference than it's counterpart. [25]

III. CONCLUSION

Machine learning algorithms provide techniques to deal with many text mining issues and this is a very interesting field of study for a researcher. Some major research issues which employ machine learning algorithms are

- **High dimensionality of textual data**
- **Sparseness of textual data**
- **Feature selection**
- **Stemming and Stop word removal**

Here is a tabular representation of the issues discussed and the algorithms used to tackle them.

Text mining issue	Algorithm used	Ideal scenario for the algorithm	Comments
High dimensionality of text data	SVD	Classic text mining problem with moderate degree of dimensionality	When computational efficiency is concerned Centroid and least square method is more effective than SVD
	Centroid and least square	Prior information of cluster structure of data is known	
	Random projection	Very large number of dimensions due to which applying classical methods would be impractical	
Sparseness of text data	PCA and sparse PCA	Sparcity of the corpora is between 95 to 99% and number of feature is greater than number of samples	sparse PCA is computationally less expensive than traditional PCA
Feature selection	Gini index	Discriminating factors of the terms need to be analyzed	
	Entropy based methods		
	Chi square and MI		
Classifier	SVM	General text categorization tasks	Choosing a classifier depends on the nature of the task in hand
	ANN	Supervised text categorization tasks	
	Decision tree based	Decision support tools with non-expert users	
Stemming	Truncating methods	Fast execution is needed	
	Statistical methods	Language independent stemmer is to be used	

As statistical algorithms and machine learning concepts are often used while solving various text mining problems, these fields of study share their common issues. If a text mining task can not be done effectively by a specific algorithm, we may look for alternative approaches either in statistical methods used to solve the problem or in the underlying machine learning concepts. As research in text mining progresses, new alternative methods and techniques to explore the field will emerge.

REFERENCES

[1] Ron Bekkerman, Ran El-Yaniv, Naftali Tishby, and Yoad Winter. On feature distributional clustering for text categorization. In Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, pages 146–153. ACM, 2001.

[2] A Anil Kumar and S Chandrasekhar. Text data pre-processing and dimensionality reduction techniques for document clustering. In International Journal of Engineering Research and Technology, volume 1. ESRSA Publications, 2012.

[3] Baharum Baharudin, Lam Hong Lee, and Khairullah Khan. A review of machine learning algorithms for text-documents classification. Journal of advances in information technology, 1(1):4–20, 2010.

[4] Vandana Korde and C Namrata Mahender. Text classification and classifiers: A survey. International Journal of Artificial Intelligence & Applications (IJAIA), 3(2):85–99, 2012.

[5] Youngjoong Ko and Jungyun Seo. Automatic text categorization by unsupervised learning. In Proceedings of the 18th conference on Computational linguistics-Volume 1, pages 453–459. Association for Computational Linguistics, 2000.

[6] Haesun Park, Moongu Jeon, and J Ben Rosen. Lower dimensional representation of text data based on centroids and least squares. BIT Numerical mathematics, 43(2):427–448, 2003.

[7] Charu C Aggarwal and ChengXiang Zhai. Mining text data. Springer Science & Business Media, 2012.

[8] Inderjit S Dhillon, Yuqiang Guan, and Jacob Kogan. Iterative clustering of high dimensional text data augmented by local search. In Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on, pages 131–138. IEEE, 2002.

[9] Ella Bingham and Heikki Mannila. Random projection in dimensionality reduction: applications to image and text data. In Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, pages 245–250. ACM, 2001.

[10] Inderjit S Dhillon and Dharmendra S Modha. Concept decompositions for large sparse text data using clustering. Machine learning, 42(1-2):143–175, 2001.

[11] Youwei Zhang and Laurent E Ghaoui. Large-scale sparse principal component analysis with application to text data. In Advances in Neural Information Processing Systems, pages 532–539, 2011.

[12] Sanasam Ranbir Singh, Hema A Murthy, and Timothy A Gonsalves. Feature selection for text classification based on gini coefficient of inequality. In FSDM, pages 76–85. Citeseer, 2010.

[13] Christine Largeron , Christophe Moulin , and Mathias G´ery . Entropy based feature selection for text categorization. In Proceedings of the 2011 ACM Symposium on Applied Computing, pages 924–928. ACM, 2011.

[14] Phayung Meesad, Pudsadee Boonrawd, and Vatinee Nuipian. A chi-square-test for word importance

differentiation in text classification. Proceedings of Computer Science and Information Technology, 6:110–114, 2011.

[15] Charu C Aggarwal and Nan Li. On node classification in dynamic content-based networks. In SDM, pages 355–366. Citeseer, 2011.

[16] Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. Springer, 1998.

[17] Jyrki Kivinen and Manfred K Warmuth. The perceptron algorithm vs. winnow: linear vs. logarithmic mistake bounds when few input variables are relevant. In Proceedings of the eighth annual conference on Computational learning theory, pages 289–296. ACM, 1995.

[18] Vikas Sindhwani and S Sathiya Keerthi. Large scale semi-supervised linear svms. In Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, pages 477–484. ACM, 2006.

[19] Miguel E Ruiz and Padmini Srinivasan. Automatic text categorization using neural networks. In Proceedings of the 8th ASIS SIG/CR Workshop on Classification Research, pages 59–72, 1998.

[20] J. Ross Quinlan. Induction of decision trees. Machine learning, 1(1):81–106, 1986.

[21] David E. Johnson, Frank J. Oles, Tong Zhang, and Thilo Goetz. A decision-tree-based symbolic rule induction system for text categorization. IBM Systems Journal, 41(3):428–437, 2002.

[22] Ralitsa Angelova and Gerhard Weikum. Graph-based text classification: learn from your neighbors. In Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, pages 485–492. ACM, 2006.

[23] C Ramasubramanian and R Ramya. Effective pre-processing activities in text mining using improved porters stemming algorithm. International Journal of Advanced Research in Computer and Communication Engineering, 2(12), 2013.

[24] Anjali Ganesh Jivani et al. A comparative study of stemming algorithms. Int. J. Comp. Tech. Appl, 2(6):1930–1938, 2011.

[25] Peg Howland, Moongu Jeon, and Haesun Park. Structure preserving dimension reduction for clustered text data based on the generalized singular value decomposition. SIAM Journal on Matrix Analysis and Applications, 25(1):165–179, 2003.