Survey Paper on Feature Extraction Methods in Text Categorization

Dixa Saxena
Department of Computer Science & Engineering
MANIT, Bhopal

S. K. Saritha, PhD
Department of Computer Science & Engineering
MANIT, Bhopal

K. N. S. S. V. Prasad
Department of Computer Science & Engineering
MANIT, Bhopal

ABSTRACT
As the world is moving towards globalization, digitization of text has been escalating a lot and the need to organize, categorize and classify text has become obligatory. Disorganization or little categorization and sorting of text may result in dawdling response time of information retrieval. There has been the 'curse of dimensionality' (as termed by Bellman)[1] problem, namely the inherent sparsity of high dimensional spaces. Thus, the search for a possible presence of some unspecified structure in such a high dimensional space can be difficult. This is the task of feature reduction methods. They obtain the most relevant information from the original data and represent the information in a lower dimensionality space.

In this paper, all the applied methods on feature extraction on text categorization from the traditional bag-of-words model approach to the unconventional neural networks are discussed.

General Terms
Text mining, feature extraction, neural networks, deep learning

Keywords
Bag of words algorithm

1. INTRODUCTION
In the present scenario which is surely an internet era, the importance of text categorization is increased as the number of digital documents is increasing. Its applications in real world include news stories organization, patient report in healthcare and spam filtering etc. Text categorization is actually the task of assigning predefined categories to free text documents. Thus, due to the increase in large amount of text data there is a need of efficient Natural Processing Technique to deal with large text data sets. To categorize any document, the preliminary condition is to find the optimal number of attributes or features which is known as dimensionality reduction. As the size of the text will increase, redundancy or noise data will increase as well. There are two broad ways to do this task: (i) selecting a subset of the original feature set which are generally discriminative i.e. feature selection (FS) [2] or (ii) building a new set of features from the original feature set i.e. feature extraction (FE) [3]. As the extraction is seen to outperform the selection techniques, only extraction techniques will be discussed further.

The main tasks of feature extraction methods are (i) to obtain the most relevant information from the original data and represent the information in a lower dimensionality space, and (ii) constructing combinations of the variables to get reduced number of features while still describing the data with sufficient accuracy.

The text data directly cannot be fed in the machine for extracting features, as most of the algorithms expect the feature vectors of the text as input. Thus, before feeding in actual machine, the raw data has to be represented in one of the following forms:

1.1 Bag of words representation [4]: Under this form, every sentence in the document is considered to be a multi-set or bag of words (or tokens) without considering the grammar or even the word order in it. Here, the occurrence or frequency of the word collectively contributes in features for further classification. For eg. Consider the following two documents containing sentences as below:

D1: *Meera likes dancing a lot.*

D2: John *too likes dancing but not that much*.

For the above documents, one combined list is made:

[*"Meera", "likes", "dancing", "a, "lot", "John", "too", "but", "not", "that", "much"*]

1.2 Vector Space Model [5,10]: This is an algebraic model for text representation. It consists of three stages:

1.2.1 Stage 1: Indexing of the documents where the content bearing terms [6] are extracted from the document text. The terms having very high or very low frequency distract the learning and hence are eliminated. Such words are known as function words [6,7,8]. These include the highly occurring stop words like *"a, an, the, on".* For eg:

"New York** is using **sand-filled trucks** to **protect Thanks giving parade".

Here, the words in bold are the content bearing words.

1.2.2 Stage 2: Weighting of the indexed terms for the enhancement of the retrieval of relevant document. There are many ways to give weight to the terms depending upon the application.

$D_j = (w_{1,j}, w_{2,j} \ldots w_{n,j})$ is the representation of document in terms of weights.

Here, each dimension corresponds to an independent term. Zero shows absence of any term from the document. The most

common method to calculate these weight terms is to use the frequency of the word as its weight. But while calculating precision and recall, it was seen that the frequency has an inversely proportional relation with its importance in the document. Hence TF-IDF (term frequency-inverse document frequency) came out as the best possible solution.

$$\text{Tf-idf}_{t,d} = \text{tf}_{t,d} \times \text{idf}_t$$

Where,

$\text{Tf}_{t,d}$ is the term frequency of term 't' in document 'd' of corpus 'D'. i.e.

$$\text{Tf}_{t,d} = 0.5+0.5\times \frac{ft,d}{\max\{ft',d: t'\epsilon d\}}$$

idf_t is the inverse document frequency which tells how much a term can provide information .i.e.

$$\text{Idf}_{t,D} = \log \frac{N}{|\{d\epsilon D: t\epsilon d\}|}$$

1.2.3 Stage 3: Ranking of the documents taking the similarity measure into consideration to get the closet words from query document. The most popular similarity measure is the cosine coefficient, which measures the angle between the document and query vector i.e.

$$\text{Cos } \theta = \frac{d.q}{||d||\,||q||}$$

Where,
d.q is the dot product of the document and the query vectors. ||d|| is the normalized form of d, ||q|| is the normalized form of vector q. the normalized form of any vector can be calculated as:

$$||q|| = \sqrt{\sum_{i=1}^{n} q2}$$

Other methods include jaccard coefficient and Dice coefficient [9].

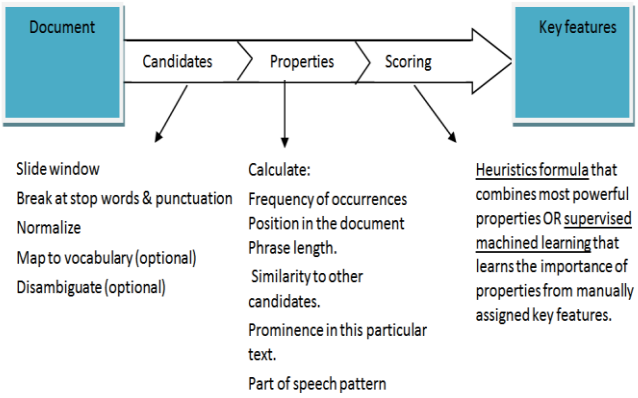

Figure 1 explains how a typical vector space model algorithm works.

Rest of the paper is divided in the following sections. Section II contains literature survey where various methods to extract significant features from text which includes the traditional as well as the unconventional methods are conversed, Section III contains the research gap that is found in the various approaches discussed, and Section IV contains the conclusion for all the survey done.

2. LITERATURE SURVEY

For conventional text clustering each document is represented as a vector using Feature Vector Model (FVM) also known as Vector Space Model (VSM) [5,10]. But a widely used approach for document representation is bag of words and the algorithms that use this representation are bag of words algorithms. Some of the methods practiced for the same are described below:

2.1 PCA- Principle Component Analysis

PCA is one of the **best known methods** that is used for feature extraction. PCA is a method used for the linear transformation of feature vector. Projection of d dimension data into m dimension Feature space, d>m such that

a. Variance of projected data is minimized.
b. Mean square error is minimized.

In text clustering, dimension reduction is one of the prominent pre-processing steps. Various FS/FE methods and two-stage models based on FS methods and/or FE methods are already proposed and implemented in the literature for dimension reduction. In case of one-stage model, dimension reduction can be performed either with a FS method [11,12,13] or with a FE method [14,15]. Removal of irrelevant, redundant and noisy features which affect performance of the underlying algorithm negatively, is not possible with one-stage dimension reduction methods [16,17,18,19,20]. As the FE methods preserve most of the information of the original feature space while reducing the dimension, their performance is significantly affected by the irrelevant and redundant features present in the original space. For this reason they are not a good choice in case of presence of high degree of irrelevant and redundant features in the feature space. On the other hand, FS methods are better choice to remove irrelevant and redundant features. Though FS methods efficiently identify relevant features in the original feature space without getting much affected by the irrelevant features, they fail to retain information in the original space. It suggests that hybrid model is better way to utilize the advantages of one method to lessen the drawback of the other method to achieve (near) global optimal solution [26].

Therefore, recently two-stage models have been proposed by various researchers [21,22,23,24,25]. In this case, selection of discriminative features is performed with two different feature scoring methods having different characteristics. This methodology provides an additional advantage to deal with features having different types of problem characteristics together, hence construct a more informative space compared to the one-stage model. There are four possible ways FS-FS, FS-FE, FE-FS, and FE-FE to conduct two-stage dimension reduction. Out of all the above combinations FS-FS and FS-FE are used for dimension reduction as the other options are not suitable for dimension reduction.

However, despite various advantages, two-stage dimension reduction methods suffer from some inherent drawbacks. Two-stage dimension reduction FS-FS methods remove irrelevant and redundant features only. Though FS-FS methods reduce dimensions in the feature space significantly, they fail to remove noisy features along with preserving valuable information present in feature space. On the other hand, FS-FE methods first use FS methods to remove irrelevant features and then use FE method to transform high dimensional feature

space into low dimensional feature space, which contains as much of the information as possible in original feature space. However, transformation of high dimensional feature space into low dimensional feature space is performed in redundant feature space, which affects the transformed feature space significantly.

So, [Kusum Kumari Bharti et al, 2014] [26] proposed a framework for three-stage dimension reduction models that combine the strengths of two FS methods and one FE method to create relevant, non-redundant and noiseless features subspace along with preserving valuable information. The performance of the proposed models is compared with one-stage models and two-stage models with respect to dimension reduction rate, clustering accuracy, and execution time. Table 1 mentions the summary of the approaches used by the author.

Datasets	One-stage versus three-stage	Two-stage versus three-stage	Remarks
N(NOD)	(7863-430)+	(7623-430)+	
		(420-430)+	
	(8278-438)+	(7987-438)+	
		(425-438)+	
R(NOD)	(3972-417)+	(3891-417)+	
		(409-417)-	
	(4234-433)+	(4234-433)+	Reduction in number of dimensions, smaller values are preferable
		(419-433)-	
W(NOD)	(3506-336)+	(3506-336)+	
		(330-336)-	
	(3670-343)+	(3550-343)+	
		(335-343)+	

It is due to the fact that the first FS method in the proposed three-stage FS-FS-FE model ie. MAD-AC-PCA (Mean absolute difference, absolute cosine, and principal component analysis) effectively obtains the relevant features, the second FS method removes the redundant features, and at last the FE method efficiently reduces the dimension (features) and noise along with preserving valuable information. Therefore, the proposed three-stage models where specifically PCA is added are effective and efficient to remove irrelevant, redundant and noisy features along with preserving valuable information in the high dimensional datasets.

The introduction of Principal component analysis (PCA) [27] as a third stage made all the difference. It was firstly introduced by Karl Pearson in 1901. It is also known as Karhunen–Loeve or K–L method. It is a statistical method that uses an orthogonal linear transformation to transform a high dimensional feature space into a new low dimensional feature subspace. The number of transformed principal components may be less than or equal to the original variables. Let m be the number of original variables and p be the number of transformed variables, then p may be less than or equal to m ($p \leq m$). Transformation of data to a new space is carried out in such a way that the greatest variance of the data by any projection lies on the first component (called the first principal component), the second greatest variance lies on the second component, and so on. In other words, each component is having high variance than its succeeding component and less variance then the preceding component (refer [28], for a detailed description of mathematical process of PCA). It has several advantages, e.g., it is computationally inexpensive, it can handle sparse and skewed data, it has an ability to remove noisy features.

2.2 Artificial Neural Network

[Jian chang Mao and Ani1 K. Jain, 1995][29] proposed a number of networks and learning algorithms which provide new or alternative tools for feature extraction and data projection. These networks include a network (SAMANN) for Sammon's nonlinear projection, a linear discriminant analysis (LDA) network, a nonlinear discriminant analysis (NDA) network, and a network for nonlinear projection (NP-SOM) based on Kohonen's self-organizing map [30,31,32]. They evaluated five representative neural networks for feature extraction and data projection based on a visual judgment of the two-dimensional projection maps and three quantitative criteria on eight data sets with various properties. Sammon [33] proposed a nonlinear projection technique that attempts to maximally preserve all the inter pattern distances using the following weight updation formula:

$$\Delta w_{jk}^{(L)} = -\eta \frac{\partial E_{\mu\nu}}{\partial w_{jk}^{(L)}}$$

$$= -\eta(\Delta_{jk}^{(L)}(\mu) y_j^{L-1}(\mu) - \Delta_{jk}^{(L)}(\nu) y_j^{L-1}(\nu)) \quad \ldots\ldots (1)$$

where η is the learning rate.

Similarly, the general updating rule for all the hidden layers can be obtained $l = 1. \ldots . L-1$ using Eq(2):

$$\Delta w_{ij}^{(l)} = -\eta \frac{\partial E_{\mu\nu}}{\partial w_{ij}^{(l)}}$$

$$= -\eta\left(\Delta_{ij}^{(l)}(\mu) y_i^{(l-1)}(\mu) - \Delta_{ij}^{(l)}(\nu) y_i^{(l-1)}(\nu)\right) \quad \ldots\ldots\ldots (2)$$

Where,

$$\Delta_{ij}^{(l)}(\mu) = \delta_j^{(l)}(\mu)[1 - y_j^{(l)}(\mu)] y_j^{(l)}(\mu)$$

$$\Delta_{ij}^{(l)}(\nu) = \delta_j^{(l)}(\nu)[1 - y_j^{(l)}(\nu)] y_j^{(l)}(\nu)$$

And

$$\delta_j^{(l)}(\mu) = \sum_{k=1}^{m} \Delta_{jk}^{(l+1)}(\mu)\, w_{jk}^{(l+1)}$$

$$\delta_j^{(l)}(\nu) = \sum_{k=1}^{m} \Delta_{jk}^{(l+1)}(\nu)\, w_{jk}^{(l+1)}$$

SAMANN Unsupervised Back propagation Algorithm:

1) Initialize weights randomly in the SAMANN network.

2) Select a pair of patterns randomly, presents them to the network one at a time, and evaluates the network in a feed forward fashion.

3) Update the weights using eq (1) and eq (2) in the back propagation fashion starting from the output layer.

4) Repeat steps 2-3 a number of times.

5) Present all the patterns and evaluate the outputs of the network; compute Sammon's stress; if the value of Sammon's stress is below a pre-specified threshold or the number of iterations (from steps 2-5) exceeds the pre-specified maximum number, then stop; otherwise, go to step 2.

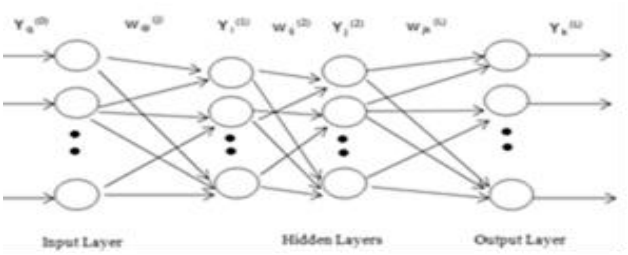

Figure 2

Figure 2 shows the three-layer feed forward network for Sammon's projection (SAMANN) and nonlinear discriminant analysis (NDA**).**

2.3 Neuro-Fuzzy Method

[Rajat K. Dea, Jayanta Basakb, 2001][34] demonstrated a way of formulating a neuro-fuzzy approach for feature extraction under unsupervised training. A fuzzy feature evaluation index for a set of features is newly defined in terms of degree of similarity between two patterns in both the original and transformed feature spaces. A layered network is designed for performing the task of minimization of the evaluation index through unsupervised learning process. This extracts a set of optimum transformed features, by projecting n-dimensional original space directly to n-dimensional (n:n) transformed space, along with their relative importance. This method gave better results than PCA[27]. Figure 3 shows a schematic diagram of the proposed neural network model.[34]

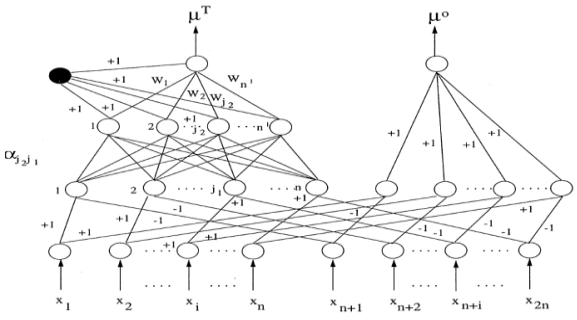

Figure 3[34]

The network (Fig. 3) consists of an input, two hidden and an output layers. The input layer consists of a pair of nodes corresponding to each feature. The first hidden layer consists of 2n(for n dimensional original feature space) number of nodes. Each of the first n nodes computes the part X_i of Eq. (3) ie:

$$d_{pq} = [\sum_j w_j^2 (\sum_i x_{ji}(x_{pi} - x_{qi}))^2]^{1/2}$$

$$= [\sum_j w_j^2 (\sum_i x_{ji}(^{x}_{i}))^2]^{1/2} \quad , \; ^{x}_{i} = x_{pi} - x_{qi}$$

$$= [\sum_j w_j^2 \Psi_j^2]^{1/2} \; , \Psi_j = \sum_i x_{ji}(x_{pi} - x_{qi}) \ldots\ldots\ldots\ldots (3)$$

and the rest compute X_i^2 .The value of (xmax i − xmin i) is stored in each of the first n nodes. The number of nodes in the second hidden layer is taken as *n'* in order to extract *n'* number of features. Each of these nodes has two parts; one of which computes Ψ^2 of Eq. (3) and the other Φ^2 j of Eq. (4) ie.

$$d_{max} = [\sum_j (\sum_i |x_{ji}|(x_{maxi} - x_{min\,i}))^2]^{1/2}$$

$$= [\sum_j \Phi_j^2]^{1/2} \; , \Phi j = \sum_i |x_{ji}|(x_{maxi} - x_{min\,i}) \ldots\ldots (4)$$

The output layer consists of two nodes which compute μ^T and μ^o values. There is a node (represented by black circle) in between the output node computing μ^T-values and the second hidden layer. This node computes dmax (Eq. (4)) in the transformed feature space and sends it to the output node for computing μ^T. The value of β is stored in both the output nodes.

2.4 Convolutional Neural Network

Convolutions are great for extracting features from dataset. Convolutional Neural Networks (CNN) are biologically-inspired variants of MLPs. CNNs are networks composed of several layers of convolutions with nonlinear activation functions like ReLU or tanh applied to the results. Traditional Layers are fully connected, instead CNN use local connections. Each layer applies different filters (thousands) like the ones showed above, and combines their results.

[Xiang Zhang ,Junbo Zhao, YannLeCun, 2015] [35] offers an empirical exploration on the use of character-level convolutional networks (ConvNets) for text classification. They constructed several largescale datasets to show that character-level convolutional networks could achieve state-of-the-art or competitive results. Comparisons are offered against traditional models such as bag of words, n-grams and their TFIDF variants, and deep learning models such as word-based ConvNets and recurrent neural networks.

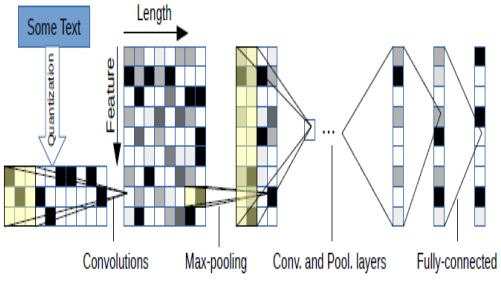

Figure 4 Illustrates the Proposed model by the author [35]

They chose two simple and representative models for comparison, in which one is word-based ConvNet and the other a simple long-short term memory (LSTM) [36] recurrent neural network model**.**

2.4.1 Word-based ConvNets: Among the large number of recent works on word-based ConvNets for text classification, one

of the differences is the choice of using pre-trained or end-to-end learned word representations. [35] offered comparisons with both using the pre-trained word2vec [37] embedding [38] and using lookup tables [39]. The embedding size is 300 in both cases, in the same way as our bag-of-means model. To ensure fair comparison, the models for each case are of the same size as our character-level ConvNets, in terms of both the number of layers and each layer's output size. Experiments using a thesaurus for data augmentation are also conducted.

2.4.2 Long-short term memory: They also offer a comparison with a recurrent neural network model, namely long-short term memory (LSTM) [36]. The LSTM model used in our case is word-based, using pre-trained word2vec embedding of size 300 as in previous models. The model is formed by taking mean of the outputs of all LSTM cells to form a feature vector, and then using multinomial logistic regression on this feature vector. The output dimension is 512. The variant of LSTM [35] used is the common "vanilla" architecture [37, 38]. They also used gradient clipping [42] in which the gradient norm is limited to 5. Figure 5 gives an illustration.

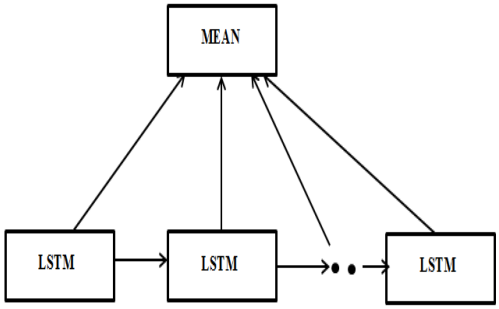

Figure 5

2.5 Deep Belief Network

It's neural network with lots of hidden layers (hundreds). State of the art for machine translation, facial recognition, text classification, speech recognition Tasks with real deep structure that humans do automatically but computers struggle with. Deep learning can be used to make multiple layers with words (letters -> words -> phrases -> sentences -> …)

[Xiang Zhang, YannLeCun, 2016][43] demonstrates that one can apply deep learning to text understanding from characterlevel inputs all the way up to abstract text concepts, using temporal convolutional networks (LeCun et al., 1998) (ConvNets). They applied ConvNets to various large-scale datasets, including ontology classification, sentiment analysis, and text categorization. They showed that temporal ConvNets can achieve astonishing performance without the knowledge of words, phrases, sentences and any other syntactic or semantic structures with regards to a human language.

3. RESEARCH GAP

While it is observed that Principle component analysis is proved to be one of the best algorithms for reducing the number of unwanted features from text and improves the accuracy by solving the redundancy problem, neural networks on the other hand is observed to be a failure when it comes to a large number of dataset. The number of neurons in input layer corresponds to the number of input words in the dataset and thus, the dataset size cannot be extended over a certain limit. Besides, it is observed that before putting the dataset on artificial neural network, it is very essential to use a feature reduction algorithm to improve the results. Thus, the neural network alone is insufficient to improve the accuracy of feature extraction.

When it comes to precision, fuzzy logic is always considered as most appropriate method. Using fuzzy with neural network one can get the results for input word lying in all categories with some probability. Hence, the result Neuro-fuzzy technique gives is better than PCA.

Whenever an artificial neural network is considered, the primary problem comes with the input number of neurons. Since there are millions of distinct words in English, the number of input neurons also reaches up to millions. The partial solution of this problem is suggested by considering the sentences as a group of characters rather than a group of words. It is known that the numbers of characters are constant in number and thus the size of input neurons is no more an issue.

The deep belief network suggests that there is no need to know the actual meaning of the word, its context or grammar. It solves all the problems that arise when one lacks in predicting the meaning of a word with respect to its use in order to categorize it.

There are many other publications mentioning different approaches for feature extraction. Table 3 below describes some of the other works done in recent past.

S.No.	Publications	Datasets used	Methodology	Results
1	[M.Ramya , J.Alwin Pinakas, 2014][44]	Selfmade, Reuters, 20newsgroup	Implemented KNN and SVM for feature selection on different datasets.	For RNN: Precision: 67.10 Recall: 66.57 F1: 66.02 For SVM: Precision: 70.74 Recall:77.24 F1: 77.96
2	[Asir Antony Gnana Singh Danasingh, Jebamalar Leavline Epiphany,2015][45]	Reuters	a term frequency (TF) with stemmer-based feature extraction algorithm(J48) is proposed	Accuracy: With stemmer: 98.5% Null stemmer: 93%

3.	[Zena M. Hira and Duncan F. Gillies, 2015][46]	-	Linear versus nonlinear classification problems, Dimensionality reduction using linear matrix factorization : projecting the data on a lower-dimensional linear subspace	
4	[Sandya H. B., Hemanth Kumar P. , Himanshi Bhudiraja, Susham K. Rao,2013][47]	Generated by using Matlab or Simulink with varying frequency in regular time interval	The Classification of extracted features is carried out by Mamdani's Fuzzy Rule based Selection System.	Out of collected 4000 sample data, 200 features were extracted.
5.	[Huan Liu, Hiroshi Motoda, Rudy Setiono, and Zheng Zhao,2010]	KDD, Ann-Thyroid,	explored the application of feature extraction on outlier detection research and proposed a novel method (DROUT) to accomplish the task.	ACP fails to preserve discriminant information, it suffers the worst performance. APCDA on the other hand slightly outperforms ERE

4. CONCLUSION

In this paper, all the possible methods that include supervised as well as unsupervised learning for extracting essential features from the text dataset are discussed and compared with the pros and cons of the approaches done before.

Also, the future gap has been discussed where it is shown that text categorization has a high scope in the field of deep learning and thus, neural networks should be applied in the field of feature extraction for text as well.

5. REFERENCES

[1] R. E. Bellman, Dynamic Programming, Princeton University Press, Princeton, NJ, USA, 1957.

[2] Isabelle Guyon, Andr´e Elisseeff, An Introduction to Variable and Feature Selection, Journal of Machine Learning Research 3 (2003) 1157-1182.

[3] PPV CJRS, Canadian Journal of Remote Sensing - 36(6):pp. 645-649; Comparison of feature extraction methods in dimensionality reduction, Electronic.

[4] Soumya George K, Shibily Joseph, Text Classification by Augmenting Bag of Words (BOW) Representation with Co-occurrence Feature, *IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 16, Issue 1, Ver. V (Jan. 2014), PP 34-38*

[5] Raghavan, V. V. and Wong, S. K. M. *A critical analysis of vector space model for information retrieval.* Journal of the American Society for Information Science, Vol.37 (5), p. 279-87, 1986.

[6] Salton, Gerard. *Introduction to Modern Information Retrieval.* McGraw-Hill, 1983.

[7] van Rijsbergen, C. J. *Information retrieval.* Butterworths, 1979.

[8] Luhn, H. P. *The Automatic Creation of Literature Abstracts.* IBM Journal of Research and Development 2 (2), p. 159-165 and 317, April 1958.

[9] Salton, Gerard. *Automatic Text Processing.* Addison-Wesley Publishing Company, 1988.

[10] A. Salton, G. Wong, C. Yang, A vector space model for automatic indexing, Communications of the ACM 18 (1975) 613–620.

[11] W. Shang, H. Huang, H. Zhu, Y. Lin, Y. Qu, Z. Wang, A novel feature selection algorithm for text categorization, Expert Systems with Applications 33 (2007) 1–5.

[12] X. He, D. Cai, P. Niyogi, Laplacian score for feature selection, in: Proceedings of Neural Information Processing Systems, 2005, pp. 505–512.

[13] Y. Li, C. Luo, S. Chung, Text clustering with feature selection by using statistical data, IEEE Transactions on Knowledge and Data Engineering 20 (2008) 641–652.

[14] X. Wang, K. Paliwal, Feature extraction and dimensionality reduction algorithms and their applications in vowel recognition, Pattern Recognition 36 (2003) 2429–2439.

[15] L. Shi, J. Zhangm, E. Liu, P. He, Text classification based on nonlinear dimensionality reduction techniques and support vector machines, in: Proceedings of the Third International Conference on Natural Computation, 2007, pp. 674–677.

[16] W. Shang, H. Huang, H. Zhu, Y. Lin, Y. Qu, Z. Wang, A novel feature selection algorithm for text categorization, Expert Systems with Applications 33 (2007) 1–5.

[17] Y. Yang, Noise reduction in a statistical approach to text categorization, in: Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'95), 1995, pp. 256–263.

[18] I. Kononenko, Estimating attributes: analysis and extensions of relief, in: Proc. European Conference on Machine Learning, Springer-Verlag, 1994, pp. 171–182).

[19] K. Kira, L. Rendell, The feature selection problem: traditional methods and a new algorithm, in: Association for the Advancement of Artificial Intelligence, AAAI Press and MIT Press, 1992, pp. 129–134.

[20] L. Liu, J. Kang, J. Yu, Z. Wang, A comparative study on unsupervised feature selection methods for text clustering, in: IEEE International Conference on Natural Language Processing and Knowledge Engineering, 2005, pp. 597–601

[21] H. Uguz, A two-stage feature selection method for text categorization by using information gain, principal component analysis and genetic algorithm, Knowledge-Based Systems 24 (2012) 1024–1032.

[22] H. Uguz, A hybrid system based on information gain and principal component analysis for the classification of transcranial doppler signals, Computer Methods and Programs in Biomedicine 107 (2011) 598–609.

[23] J. Menga, H. Lin, Y. Yu, A two-stage feature selection method for text categorization, Knowledge-Based Systems 62 (2011) 2793–2800.

[24] H. Hsu, C. Hsieh, M. Lu, Hybrid feature selection by combining filters and wrappers, Expert Systems with Applications 38 (2011) 8144–8150.

[25] A. Akadi, A. Amine, A. Ouardighi, D. About ajdine, A two-stage gene selection scheme utilizing MRMR filter and GA wrapper, Knowledge and Information System 26 (2011) 487–500.

[26] Kusum Kumari Bharti∗, P.K. Singh, A three-stage unsupervised dimension reduction method for text clustering, Journal of Computational Science,2013

[27] K. Pearson, On lines and planes of closest filt to systems of points in space, Philosophical Magazine 1 (1901) 559–572.

[28] J. Shlens, A tutorial on principal component analysis, Systems Neurobiology Laboratory, University of California at San Diego, 2005.

[29] Jianchang Mao and Ani1 K. Jain, Artificial Neural Networks for Feature Extraction and Multivariate Data Projection, IEEE TRANSACTIONS ON NELRAL NETWORKS. VOL 6. NO. 2. MARCH 1995.

[30] R. L. Hoffman and A. K. Jain, "Segmentation and Classification of Range Images," *IEEE Trans. Part. Anal. Mach. Intell.*. vol. PAMI-9. no. 5, G. 608220, 1987.

[31] K. Hornik and C.-M. Kuan, "Convergence analysis of local feature extraction algorithm," *Neural Networks,* vol. 5, pp. 229-240, 1992.

[32] W. Y. Huang and R. P. Lippmann, Comparisons between neural net and traditional classifiers," in *IEEE 1st Int. Conf. Neural Networks.* San Diego, CA, June 1987, pp. IV-485-IV-493.

[33] P. J. Huber, "Projection pursuit," *Ann. Statist.*. vol. 13, pp. 435475, 1985.

[34] Rajat K. De, Jayanta Basak, Sankar K. Pal, Unsupervised feature extraction using neuro-fuzzy approach, Fuzzy Sets and Systems 126 (2002) 277–291.

[35] Xiang Zhang, Junbo Zhao, Yann LeCun, Character-level Convolutional Networks for Text Classification.

[36] S. Hochreiter and J. Schmidhuber. Long short-term memory. Neural Comput., 9(8):1735–1780, Nov. 1997.

[37] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, editors, Advances in Neural Information Processing Systems 26, pages 3111–3119. 2013.

[38] Y. Kim. Convolutional neural networks for sentence classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1746–1751, Doha, Qatar, October 2014, Association for Computational Linguistics.

[39] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural language processing (almost) from scratch. J. Mach. Learn. Res., 12:2493–2537, Nov. 2011.

[40] A. Graves and J. Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. Neural Networks, 18(5):602–610, 2005.

[41] K. Greff, R. K. Srivastava, J. Koutn´ık, B. R. Steunebrink, and J. Schmidhuber. LSTM: A search space odyssey. CoRR, abs/1503.04069, 2015.

[42] R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks. In ICML 2013, volume 28 of JMLR Proceedings, pages 1310–1318. JMLR.org, 2013.

[43] Xiang Zhang, Yann LeCun. Text Understanding from Scratch. arXiv 1502.01710. Datasets. Code.

[44] M.Ramya , J.Alwin Pinakas, Different Type of Feature Selection for Text Classification, *International Journal of Computer Trends and Technology (IJCTT) – volume 10 number 2 – Apr 2014.*

[45] Asir Antony Gnana Singh Danasingh, Jebamalar Leavline Epiphany, Feature Extraction for Document Classification, www.researchgate.net/publication/276950476, 2015.

[46] Zena M. Hira and Duncan F. Gillies, A Review of Feature Selection and Feature Extraction Methods Applied on Microarray Data**,** Advances in Bioinformatics Volume 2015 (2015), Article ID 198363, 13 pages.

[47] Sandya H. B., Hemanth Kumar P. , Himanshi Bhudiraja, Susham K. Rao, Fuzzy Rule Based Feature Extraction and Classification of Time Series Signal, International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-3, Issue-2, May 2013.

[48] Hoang Vu Nguyen, Vivekanand Gopalkrishnan, *Feature Extraction for Outlier Detection in High-Dimensional Spaces*, Journal of Machine Learning Research, Volume 10, Issue 2, 2010, pp. 252-262