Semi-Supervised HSK-means algorithm for Contextual Document Classification

UpasanaPandey, ApoorvArora, HarshitSyal, S.Chakraverty

NetajiSubhas Institute of Technology, New Delhi

Coe.upasana@gmail.com, apoorv.1190@gmail.com, harshit.syal@nsitonline.in, apmahs@rediffmail.com

***Abstract*—Supervised learning needs a lot of labeled data to generate hypothesis function and classify test documents efficiently. In real world situations, we have a lot of unlabeled data which cannot be used in supervised learning. Hence, we propose a novel scheme for a semi-supervised learning algorithm called Half Supervised K-means algorithm. In this scheme, we input category keyword lists generated by an ontology containing lexical relations, some labeled data and test documents for classification. We have used a modified tf-idf for computing weights of keywords, labeled documents and unlabeled documents. We supply these weights to the HSK means to categories documents to their respective categories. In HSK means, the centroids are dependent on number of categories decided by users and labeled documents help to assign a category to cluster. So all documents present in same cluster automatically assigned to the category.**

***Keywords*—HSK-means algorithm; K-means clustering algorithm; semi-supervised learning; labelled data**

I. INTRODUCTION

Document classification is a task to “manually" (or "intellectually") or algorithmically assign a document to one or more classes or categories [1].This generally involves clustering a group of similar documents and assigning that cluster to a relevant category. In the *clustering problem*, we are given an *unlabeled* dataset and we need to have an algorithm that automatically groups the data into coherent clusters for classification. One of the simplest procedures to achieve this task is **K-means clustering algorithm**. To increase the power of K-means, this paper proposes a new clustering approach using advanced K-means algorithm [2] called **HSK-means algorithm**.

The K-means clustering algorithm is an iterative algorithm to cluster objects based on attributes into K partitions. It takes into account the similarity of documents within a cluster. K-means algorithm works in two steps: cluster assignment step; and move centroid step.

In “cluster assignment step”, the usual procedure is to randomly initialize n points called cluster centroids. Then the distance of every documentfrom all centroids are computed. The document, whose distance from a centroid is at minimum distance, is allocated to that centroid.

The next step is the “move centroid” step. In this step,average of values of every document in a cluster is computed. After computing the average values for every cluster separately, a new centroid is assigned at that value and the above steps are repeated until the centroids stop moving.

K-means is a simple algorithm that has been adapted to many problem domains. Due to its simplicity, it is used very frequently.

But it has some weakness also.K-means algorithm requires initialization of centroids randomly. This randomization behavior of K-means sometimes may lead it to get stuck in a bad local minimum [3], and thus gives incorrect results. To overcome with this problem, the usual procedure is to run K-means multiple times and out of all different clusters, pick the one that gives the lowest distortion J(c, μ). It also requires finding the number of centroids. This can be done manually when the clusters are well separated. In the case, when the data is continuous or not well separated, then it is hard to visualize the number of clusters.

The accuracy for correctly classifying a document amongst the above factors still depends upon the number of features. Accuracy varies as inversely proportional to the number of features. Hence a large number of features might degrade the performance of K-means. The usual procedure to handle this task is to have a feature selection method that reduces the effective features without affecting the performance. But its correct implementation is still on the charts.

Besides these pitfalls, K-means is capable to classify test documents correctly to its most appropriate category.K-means is an extremely relevant for incorporating large set of documents with such a smaller set of features which is essential for time efficiency. Due to its simplicity, it can be easily implemented.

We have considered all drawbacks and strength of K-means algorithm and propose a novel approach known as **HSK-means clustering algorithm,** which may capable to produce high accurate results for context document classification.

In HSK-means algorithm, we *do not* randomly initialize the values of the centroid. We supply category keywords list,testdocuments and some labeled documents to the algorithm. This is in contrast to the usual K-means where we just give the test documents to the clustering algorithm. In this modified algorithm, the initial centroids are allocated to the categories, not to any random document or value like in usual K-means algorithm.

In terms of performance, the usual K-means algorithm is not guaranteed to return a global optimum. As the initial centroid is randomly initialized, it may not be possible to converge at global optimum always. But as in HSK-means algorithm, because we set initial centroid to the categories only, our result is much closer to the global optimum. This algorithm converges much faster than the K-means algorithm. Hence, it is time efficient also.

It is called half-supervised because we give some *labeled* data to the algorithm. The intuition behind giving *labeled* data is that we can assign a category to a cluster, if we know that the cluster contains the *labeled* data. We don’t know which category belongs to a cluster in the K-means algorithm. But in HSK-means algorithm, the *labeled* document will specify the category to a cluster containing that document. Although it is not necessary to give the labeled data; the results can be computed efficiently even without the labeled data. So this choice is left to the user.

Another improvement can be donein the usual K-means algorithm. If a term in a document is not present in any other document or category keyword list, then that term is deleted from its document. It can be generalized as follows: If a term in a document has a df(number of documents containing the term) value less than 20 percent of the total number of documents, then that term is neglected before computing the tf-idf values. This is in contrast to supervised learning where such types of words are given higher priority for the documents containing them. This step is done to ensure the similarity of documents within a cluster. If the term is not present in any other document in a cluster then that term has no logical significance in the cluster, hence it can be removed.

Another main drawback of the K-means algorithm is to necessarily supply the number of clusters (i.e. *k*) to find. If the data is not naturally clustered, we may get strange results. Our algorithm is specifically meant for document classification or any other application where we are given the number of clusters. Hence, in our algorithm we propose to use the value for the number of clusters equal to the value of the initial categories entered by the user.

The background and motivation behind our algorithm is given in Section II. The proposed HSK-means algorithm is presented in section III. The comparisons of our algorithm with some previous approaches are given in Section IV. The conclusions and future work is discussed in section V. In brief our algorithm seems to be more accurate than usual K-means algorithm and it converges much faster than the ancient K-means algorithm. Therefore we conclude that HSK-means algorithm may highly relevant for the categorization of large corpora of text documents rather than the usual K-means algorithm.

II. BACKGROUND AND MOTIVATION

Text Categorization (or Text Classification) is a useful technique which is getting importance in these recent years. It is a process of assigning a document to a specific category or query using supervised learning. At present a number of text categorization techniques have been developed including Nearest Neighbor, Naïve Bayes, and Neural Networks and so on. However, all of these techniques require sufficient training data, hence are suitable for *supervised learning* [4].

With such an increase in the textual content over the World Wide Web, complex classification techniques are getting deployed to get efficient results. Considering the application for categorizing e-mail messages, that currently involves manually creating folders in users' mailbox and setting up rules to dispatch incoming e-mails. As a result, heavy cognitive load for creating the folder structure and the rules are required, which is generally difficult for the normal users to understand [4]. Hence, with such an increase in the number of texts over the internet or applications like email, spam filtering, etc., text classification is growing in its importance.

When statistical approaches are used for classification purposes, they obviate the need to analyze the contextual relevance of words in a document. Such procedures have the advantage of having good recall rate, but the probability of having a low accuracy and precision rate is also high. Therefore, analyzing mails with their contextual meaning should be followed as the general approach for classification. Context based classification methods essentially identify and make use of word association information to improve the classification effectiveness. Allowing the context of a word to influence its presence or absence contributes to the classification outcome.

Text clustering is important step in classification problems. The most important unsupervised learning problem, i.e., clustering, can be used to structure large sets of text or hypertext documents. The goal of clustering is to determine an intrinsic grouping in a set of *unlabeled* data. The usual approach for classification of documents begins through the pre-processing of the documents to be classified. This pre-processing component provides functions to transform real text data in different formats into vector representations in the VSM data model, so that a clustering algorithm can be applied. The functions include parsers to parse real text data in plain text, HTML, XML, PDF, PS, Word into a set of terms, stop word removal, word stemming, term selection, tokenization and term scoring such as tf-idf. Many toolkits are also available to implement these pre-processing functions, e.g., BOW toolkit is currently being deployed for pre-processing. The weights of the pre-processed vectors are then calculated to generate a tf-idf matrix, i.e., term frequency-inverse document frequency. This is given by the formula:

$tf\text{-}idf\ (d_j,t_i)=tf(d_f,t_i)*\log(|D|/df(ti))$

where $tf(dj,ti)$ = frequency of term ti in document dj,

$|D|$ = total number of documents

$df(ti)$ = number of documents in which ti occurs.

Amongst a large number of applications of clustering, one of them is K-means. K-means is one of the simplest unsupervised learning algorithms that solve the clustering problem. The procedure is to classify a given data set through a certain number of clusters (assume K clusters). The main motive is to define K centroids, one for each cluster. The centroids are placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no point is pending, the first step is completed. At this point, we need to re-calculate K new centroids as barycenter of the clusters resulting from the previous step. After we have these K new centroids, a new binding has to be done between the same data set points and the nearest new centroid. A loop has been generated. As a result of this loop we may notice that the K centroids change their location step by step until no more changes are done. In other words centroids do not move any more. Although it can be proved that the procedure will always terminate, the K-means algorithm does not necessarily find the most optimal configuration, corresponding to the global objective function minimum. The algorithm is also significantly sensitive to the initial randomly selected cluster centers. By running K-means algorithm multiple times may reduce this effect. But if the number of centroids in K-means algorithm is large then probably running k-means multiple times also won't help because then it will have even more chance of getting stuck in bad local minima. One method of choosing number of clusters is through the elbow method. In elbow method, a graph is plotted between the number of clusters and the distortion function. An observation to this graph is that the distortion function decreases linearly up to a point, and then slows down as we increase the number of clusters. The value of number of clusters where after which the distortion function slows down gives us an expected value of the number of clusters. But if the graph gives a continuous figure, which can happen if the features are quite large, then there is no way of choosing what could be the exact number of clusters. Hence for classification purposes K-means algorithm is losing its significance. K-means algorithm had to be told the number of clusters (i.e. *K*) to find. If the data is not naturally clustered, we may get strange results. In another algorithm, CR algorithm the procedure to compute the number of initial cluster centroids includes an array named *biaoji[][] that* stores the clustering index of each point, and the array's length is the number of points in the data set*(N)*. Then a parameter named *sum_distance*is computed.This variable corresponds to each value of the user-specified number of clusters. After these calculations, the correct numbers of clusters were estimated [2]. An SVM-based text classification with SSK-means clustering algorithm [5], *training data*, including both labeled and unlabeled data, are first clustered with guidance of the labeled data. The unlabeled data samples are then labeled based on the clusters obtained. This approach for text classification was using supervised learning. Also the initialization parameter for the centroid was randomly chosen in this approach. The motivation behind our approach comes from the fact that supervised learning needs a lot of labeled data to generate hypothesis function and classify test documents efficiently. In case of document classification we usually don't have enough training data. Hence, employing a semi-supervised approach for document classification can help for both supervised and unsupervised learning. A semi-supervised learning algorithm may include the advantages of both of the approaches and discard the disadvantages of one of them. Assembling the accuracy of supervised learning and the simplicity of unsupervised learning for unlabeled data, semi-supervised learning should be preferred for document classification. Accounting all of the shortcomings of the usual K-means algorithm, we propose a semi-supervised learning algorithm using a modified version of K-means called HSK-means clustering algorithm which focuses on document classification. Intuitively this scheme proposes to have a better performance than the usual K-means clustering algorithm.

III. PROPOSED WORK

For Classification of documents, it is necessary to consider the semantics of every document with each other. Documents having similar relationships are grouped into a cluster and then they are assigned a category [6]. The assignment of a document to a cluster depends on various factors like its tf-idf value and so on. But before all of these steps, we must make sure that we compute the tf-idf values of a proper pre-processed document. This pre-processing step ensures the removal of stop words and only stemmed word are passed to further steps.

Using the knowledge of this previous work, we propose a novel approach using modified tf-idf computation and HSK-means clustering algorithm. The process followed is divided into 7 parts:

A. Input Gathering Step:

In the first phase of our algorithm, the user inputsdesired categories. This input is independent of the documents to be classified and our classifier which classifies it accurately.

The documents to be classified along with some labeled documents are taken from 20 News Groups dataset. Including labeled data in our input allows us to overcome with one of the shortcomings of the usual K-means algorithm. This labeled data helps us in determining the category to be assigned to a cluster. After the centroids have stopped converging in the HSK-means clustering algorithm, we look at the labeled document within a cluster. As the labeled document itself knows to which category it belongs, hence the whole cluster is allocated to that category only. Although, it is not necessary for the user to supply some labeled data. The algorithm computes the results thereafter also, but with a little reduction in the accuracy level. Hence it left to the user in determining whether or not to give labeled data. The ratio between labeled and unlabeled documents must be in between 1:5 to 1:4 in order to get the effective results. This is in contrast to supervised learning where we have to provide a bunch of labeled data to train the classifier.

B. Category Keyword list generation:

In the background, as the user supply category names to the proposed algorithm, we compute a semantic keyword list using word net and 1st level reference of Wikipedia for each category. Wiktionary can also be used, as it

gives better keyword list than Word net and Wikipedia [7]. Thus we are left over with a number of lists, where each list contains keywords which reflect the lexical relationsof its respective category.
This keyword generation step is an important step because in proposed HSK-means algorithm we initialize the centroids as according to the values of this keyword list only. This improvement on the usual K-means helps overcoming the drawback of converging to local minima. Hence, the appropriate keyword list is the main concern in this step.
Also, the weights are assigned to the keywords in category keyword list. This makes use of a weight function, which gives the weight or strength of a term in a list for its respective category. This weight can be used to differentiate the importance of terms within a list for its category.

C. Pre-processing:

Ensuring proper stemmed words [8] without any stop words of the documents, is a common step for every classifier. Data in the real world may be incomplete (lacking attribute values), noisy (containing errors and outliners) and inconsistent (containing discrepancies). Hence, to ensure that only real values of the documents are passed to the classifier:

- The pre-processing of all unlabeled documents is done through a modified version of Porters-Stemming algorithm [21] which includes Stop Word Removal and Tokenization also. This pre-processing gives us the desired tokens that are used in further steps. It is based on these tokens that the tf-idf values are computed, and not the original document.
- Most of usual classifiers take into their account only pre-processed documents, but here in our algorithm, not only test documents are preprocessed, but also each category keyword list is pre-processed. This pre-processing step includes stemming and tokenization. This pre-processing of category keyword list is necessary to give satisfactory results. The intuition behind this practice is when we are looking at the context, rather than the statistical value, we must have the two words spelled equally, in order for them to be equal. So if we are pre-processing test documents, then pre-processing of category keyword list is also required in order to match tokens of document and keyword list.

Previous classification methods are notrequiring use of this step; as a result their accuracy degraded abruptly. But our algorithm ensures that a term in a document is equivalent to a keyword in the category list, if both of them are derived from the *same* word, as both are already pre-processed. Hence it gives more power to our algorithm.

D. Selective Processing:

K-means algorithm classifies documents based on the similarity between the documents. Considering all of the documents to be classified, there may be many terms that appear only in one document. Such words which appear only in one document tend to affect the similarity proportion of the document with the other documents in a cluster. Hence, if a term in a document is not present in any other document or category keyword list, i.e., if the df value (number of documents containing that term) of the term is equal to 20 percent of the value of the value of |D| (total number of documents), then that term is deleted from its document and the document is updated. This step is done to ensure the similarity of documents within a cluster. If the term is not coming in any other document or category keyword list in a cluster then that term has no logical significance in the cluster, hence it can be removed. Another goal obtained by this step to reduce tf-idf table, as resultant system has less memory requirementsandis also time efficient. Also, the overhead for computation of the distance of the term from the centroid is neglected.
The selective processing step is in contrast to supervised learning where such types of words are given higher priority for the documents containing them.

E. Advanced TF-IDF:

A tf-idf table computation is an important step for every classifier. Here the tf, which stands for the term frequency and idf which stands for inverse document frequency, is computed for every term matched with the documents. Term Frequency is given by the number of times it is appearing in the document, i.e., its frequency, and Inverse Document Frequency is given by the logarithm of the ratio between D, the total number of documents and df, the number of documents containing that term. Most of the usual classifiers compute the tf-idf values of tokens of documents, acquired after pre-processing them [9].
But in our algorithm an advanced tf-idf table is generated, which comprises of the unlabeled documents, labeled documents and category keyword list. The intuition behind the submission of pre-processed category keyword list is that it provides us the initial centroid for HSK-means algorithm. The reason for setting the parameter of initial centroids to the category keyword list is simple. Random initialization for the centroids might degrade the performance of the usual classifier, as it may get stuck in a bad local minimum. To overcome with this difficulty of usual classifiers, we set this parameter to the category keyword list, and then our algorithm will guarantee to converge at global optimum. The labeled documents help in determining the category to which a cluster belongs in the further category allotment step. The cluster containing the labeled data knows which category it is to be assigned. The aim is to know which indices are for labeled documents and which are for category keyword list.

F. HSK-means clustering algorithm:

The heart of our algorithm is the final and an improved version of the usual K-means algorithm [2]. Considering our main objective which was to ensure that every document correctly classified to the appropriate category, we propose a new clustering algorithm that not only solves the problem accurately, but also wins over the previous classifiers in terms of time and space utilization requirements.
The HSK-means algorithm *does not* randomly initialize the values of the centroid. Neglecting the randomization process in our algorithm, greatly affects the accuracy with which it may function. We supply category keywords list and test documents to HSK-means algorithm. So if we set the category keyword list only as parameter for the initial centroids, then we may not get strange results. It is also the case when the algorithm will converge to a global optimum, rather than a bad local minimum in case of the usual procedure.
Also, supplying some labeled data to our algorithm greatly enhances the performance of the algorithm. This labeled data, which was inputted at initial steps, helps the HSK-means algorithm in allocating a category to a cluster. If a cluster contains a labeled document, then the labeled document automatically assigned to its category. Hence, we can allocate that category to the overall cluster.
Many previous approaches which are discussed in the comparative study section give different ways to set the parameter of the number of centroids. But they all require some computation and hence are not efficient in terms of time and space. Our algorithm is focused on those applications where we know the number of clusters. As the aim of this proposal is document classification, hence the number of clusters is equal to the number of categories desired by the user.
Hence we set the parameters for H.S. K-means algorithm as follows:
i) The number of centroids is equal to the number of categories inputted by the user in the 1st step.
ii) Initialize cluster centroids 1, 2…K to the category keywords list prepared earlier.

1. Initialize clusters 1, 2 ….K to the category lists prepared earlier.
2. Repeat until convergence {
3. for every i in unlabeled and labeled documents
4. $c^{(i)}$[for unlabeled documents]= arg min$\|x^{(i)}- \mu_j\|^2$
5. $c^{(i)}$[for labeled documents]= μ_j(centroid belonging to the category keyword list for label in $x^{(i)}$)
6. for each j(centroid), set
7. μ_j=sumof({$c^{(i)}$=j}$x^{(i)}$)j=1 to n divided by sumof({$c^{(i)}$=j}) for j=1 to n }

Figure 1: Proposed HSK means algorithm for document classification

Hence, now clusters will be formed with each document belonging to one of the clusters.

G. Category allotment:

After clustering, we give label to the unlabeled documents according to the label of category keyword list of the centroid of that cluster.

Then we run our HSK-means algorithm for many times to ensure HSK-means is not getting stick in bad local minima. Although the probability of getting stuck in a bad local minima, is very less as compared to that of the usual K-means algorithm.

Then, out of all the different obtained clusters, pick the one that gives the lowest objective function.

The objective function

$$J=\sum_{j=1}^{k}\sum_{i=1}^{n}\|x^{(i)}-c_j\|^2$$

Where $\|x_i^{(j)}-c_j\|^2$ is a chosen distance measure between a data point $x_i^{(j)}$ and the cluster centre c_j, is an indicator of the distance of the *n* data points from their respective cluster centers.

This scheme may have better performance than K-means clustering algorithm and it also overcomes the inefficient behavior and the strange results of K-means, thereby chance of increasing the accuracy for classification.

IV. COMPARITIVE STUDY

Traditional K-means has lot of problems when classifying documents which have very high dimensional feature vectors. One way of improving the quality of K-means is to employ better annealing techniques [10] to avoid (bad) local optima [3]. Other ways of improving K-means include employing its spherical version, heuristic to change it into an iterative procedure of K-means, as done in Organizing Map [11] and Neural-Gas [12], Online Spherical K-Means [13].

Recent years have noticed a billowing interest in development of machine learning algorithms that are able to use labeled data with additional unlabeled data in classification process. Nowadays in document classification large quantities of unlabeled textual data are readily available. In general problem of using unlabeled data in supervised learning leads to a semi-supervised learning or labeled-unlabeled problem in different context.

A number of works have been reported in developing semi-supervised document classification, including Co-Training (Blum & Mitchell) [14], Transductive SVM (TSVM) (Joachims, 1999) [15], EM (Nigram) [16],SSK-Means [5] and complete review could be found in Seeger [17] and Shi Zhong [18].These proposals obtain considerable improvement over traditional supervised methods when the size of training dataset is relatively small, but they face difficulties when the labeled dataset is *extremely* small. This is somehow expected as most of those methods employ same iterative procedure which train an initial classifier intemperately based on the distribution presented in the labeled data. When labeled data is small, and the unlabeled samples are far apart from corresponding class centers due to the high dimensionality, these methods will often have a poor starting point and conglomerate more errors in further iterations [19]. Also there is error propagation phenomenon which is particularly relevant for text classification tasks because textual documents generally have very high feature dimensionality. Therefore, incorporating additional unlabeled data in model learning can actually increase or decrease classification accuracy, and the impact of unlabeled data on learning needs to be carefully modulated to make them useful [20].CBC [19] a clustering approach treats semi-supervised learning as *clustering* aided by the *labeled* data, while the other existing algorithms treated it as *classification* aided by the *unlabeled*data.

Our approach for clustering is similartoCBC,i.e. we view semi-supervised learning as clustering aided by labeled data but in our approach classification metric have been shifted from similarity in documents to similarity between category list for a category and corresponding documents.

V. CONCLUSIONS AND FUTURE WORK

HSK-means algorithm may have better performance on document classification than usual K-means algorithm. This scheme replaces the inaccurate results acquired by the usual K-means algorithm with the appropriate ones, still enhancing the efficiency. Time and space utilization is another goal that HSK-means manages to cover, which the usual K-means failed to converge. Pre-processing of category keyword list ensures that keywords match up properly with the tokens obtained through document pre-processing. Although this may require manually upgrading the keyword list at a later stage, but it can be achieved with keyword list enrichment. Selective processing task cannot be used for supervised learning, whereas it is good option for semi-supervised learning problems.

Future work includes the implementation of the proposed work along with the addition of anomaly detection, to make use of Gaussian distribution thereby enhancing the results and classifying the documents with accurate and efficient results. Also keyword list enrichment and word net disambiguation are few more issues to deal with in future scenario.

REFERENCES

[1] Wikipedia-http://en.wikipedia.org/wiki/Document_classification

[2] Bin Wang,"*A New Clustering Algorithm Compared With the Simple K-Means"*, In procs. of IEEE, 2009, Wuhan.

[3] Kenneth Rose, *"Deterministic annealing for clustering, compression, classification, regression, and related optimization problem"*, In proc of IEEE, vol. 86, no. 11, pp. 2210–2239, 1998.

[4] Yang Xiang, Wanlei Zhou, and Jinjun Chen Managing,*"Email Overload with an Automatic Nonparametric Clustering Approach"*, In procs. of International Federation for Information Processing, 2007, Ethiopia.

[5] Hongcan Yan , Chen Lin, Bicheng Li, *"A SVM-based Text Classification Method with SSK-means clustering algorithm",*In procs. of International Conference on Artificial Intelligence and Computational Intelligence, 2009, Monte Carlo Resort, Las Vegas, Nevada, USA.

[6]KazemTaghva and RushikeshVeni,*"Effects ofSimilarityMetricson Document Clustering"*, In proc of 2010 Seventh International Conference on Information Technology: New Generations, Washington, DC, USA.

[7] TorstenZesch and Christof Müller and IrynaGurevych,*"Extracting Lexical Semantic Knowledge from Wikipedia and Wiktionary"*, In proc of 6th International Conference on Language Resources and Evaluation, *May 2008.*

[8]AlgorithmsIliaSmirnov,*"Overview of Stemming"*, In proc. of DePaul University03/12/08

[9] T. Joachims,*"A Probabilistic Analysis of the Rocchio Algorithm withTFIDF for Text Categorization"*,In procs. of the 14th International Conference on Machine Learning ICML97. 1997, Nashville, Tennessee, USA.

[10] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi,*"Optimization by simulated annealing",Science*, vol. 220, pp. 671–680, 1983.

[11] TeuvoKohonen, *"The Self-Organizing Map"*, Inprocs. of the IEEE,vol. 78, no. 9, pp. 1464–1480, September 1990, Europe.

[12] T. M. Martinetz, S. G. Berkovich, and K. J. Schulten, *IEEE Trans. Neural Networks*, vol. 4, no. 4, pp. 558–569, July 1993.

[13] Shi Zhong,*"Efficient Online Spherical K-means Clustering"*, In proc of IEEE International Joint Conference on Neural Networks 2005, Portland.

[14] A. Blum and T. Mitchell, *"Combining labeled and unlabeled data with co-training, "* In Proc. of 11th COLT conference, 1998, pp. 92-100.

[15] Junhui Wang, XiaotongShen, Wei Pan, *"Transductive Support Vector Machines"*, In proc. of 5th SIAM International Conference on Data Mining, Newport Beach, California.

[16] K. Nigam, A. McCallum, S. Thurn and T. Mitchell, *"Text Classification from labeled and unlabeled documents using EM"*, In proc. of Machine Learning, 39 (2/3), 2000, pp.103-134.

[17] Seeger, M. (2001),*"Learning with labeled and unlabeled data"*, Technical report, EdinburghUniversity.

[18] Shi Zhong,*"Semi-supervised Model-based Document Clustering: A Comparative Study"*, In proc. of IEEE Int. Conf. Computer Vision Pattern Recognition,Florida.

[19] H. Zeng, X. Wang, Z. Chen, H. Lu and W. Ma, *"CBC-clustering based text classification requiring minimal labeled data"*,InProc. of 3rd International Conference on Data Mining, 2003, pp.443-450.

[20]JieJiTon,Y. T. ChanQiangfuZhao,*"Comparative Advantage Approach for Sparse Text Data Clustering"*, In proc. ofNinth IEEE International Conference on Computer and Information Technology, 2009, TBD Xiamen, China

[21] Porter M.F. *"An algorithm for suffix stripping"*, Program. 1980; 14, 130-137.