Hybrid Clustering Approach for Concept Generation

K.Thammi Reddy[†] **M.Shashi**[††] **and** **L.Pratap Reddy**[†††],

[†] GITAM College of Engineering, Rushikonda, Visakhapatnam, Andhra Pradesh, India
[††]Andhra University , Visakhapatnam, Andhra Pradesh, India
[†††]JNT University, Hyderabad, Andhra Pradesh, India

Summary

Information retrieval is one of the major research areas due to accumulation of huge information in digital form. Various techniques of Information retrieval are based on the fact that various terms present in a document along with their frequency of occurrence signify the semantics of the document. Recent attempts to find the relevant document for a context represents documents in a Latent Semantic Indexing (LSI) model as document-term vector representing term weights for every index term in that document. As there will be enormous number of index terms this leads to high dimensionality problem. We can reduce the dimensionality based on the observation that groups of terms associated with related concepts occur together or do not occur in a document based on whether the document is relevant or not to that concept. Such a group of terms is identified as a Concept and can be viewed as a single dimension in a Rough set based information retrieval system. In this paper we present a hybrid clustering approach for the formation of equivalence classes of terms associated with related concepts. It uses the outcome of hierarchical clustering to provide seed points for implementing Incremental K-means algorithm. Due to the sparsity of the term vector, the cosine similarity estimate is found to be less effective for term clustering. Another promising measure of proximity estimate generally used in information retrieval is the Euclidian distance that it is biased towards changes in the term frequencies in larger documents when the term weights are represented by Term frequency-inverse document frequency (tf-idf) estimates. In this paper we propose a new term weight estimate namely term probability–inverse document frequency (tp-idf) for representing a term as a vector before clustering the terms.

Key words:
Hierarchical clustering, Partitional clustering,
Text mining, Dimensionality reduction, Proximity estimate.

1. Introduction

The advent of Information technology in various sectors leads to accumulation of digitized information. This can be described [56.] as data rich but information poor. It is exceeding our human ability for comprehension without powerful Information retrieval systems.

In an information retrieval system document can be represented using one of the three classical models namely Boolean [2], vector [16], and probabilistic [13] LSI [3] models to retrieve the relevant documents from huge repositories. The Boolean model [1] considers the existence or non-existence of index terms in a document. As a result, the index term weights are defined with a simple binary value. The vector model uses non-binary weights to index terms indicating how frequent they are in a document or how important they are in a query. In the probabilistic model, given a user query and a document it tries to estimate the probability that the user find the document interesting. The Latent Semantic Indexing (LSI) attempts to capture the statistical relationships among terms of the vocabulary. In LSI, the document space in which each dimension is an actual term occurring in the collection is replaced by a much lower dimensional document space called k-space in which each dimension is a derived concept, a “conceptual index,” called an LSI “factor” or “feature.” Unlike the terms of the vocabulary, these LSI factors are totally independent from one another based on statistics. Out of many information retrieval systems, one mechanism used for retrieval task is the Rough Set based Information Retrieval System (RSIR) [10]. Rough set theory is an extension of conventional set theory that supports [7] approximations in decision making. It possesses many features in common to a certain extent with the Dempster-Shafer theory of mathematical evidence and Fuzzy Set theory. In the rough information retrieval system the major problem is to find an appropriate equivalence relationship for partitioning the attributes into different subsets.

In the information retrieval context there are different objects which may be partitioned [8] using equivalence relations. In one context the equivalence relation will place documents that are similar to each other in the same class using probability model. The approach adopted by Wong [14] is based on the probability model. The major drawback of this model is its computational complexity and the method is totally based on usage statistics. Other approaches partition the attributes based on the opinion of the domain expert. The above methods are laborious and lacking behind with inadequate efficient procedures to

Manuscript received April 5, 2007
Manuscript revised April 25, 2007

automate the process of equivalence class generation that play an important role in the design of a Rough Set Information Retrieval Systems.

Partitioning of documents into classes based on similarity is a subjective evaluation of similarity carried out with an estimation of independent corpus clustering and overlapping measure in multiple clusters. However, document similarity may also be computed using a variety of similarity measures based on the indexing information. Or the strategy is to group documents that are relevant to the same set of queries. In addition, citations and other such linkages are used to derive classes of similar documents. In the other context the set of objects that may be partitioned is the query set. Here the equivalence relation will be selected such that each equivalence class represents queries that are similar as determined by the terms in common or the similarity is determined as a function of the relevant documents in common. In the third context the equivalence relations are used to partition the indexing (search) vocabulary of the database. The idea here is to group together terms that refer to the same concept in an equivalence class. Although the document clustering problem is addressed extensively in [1], the advantage of clustering terms along with documents is recognized as another important issue to be considered. A recent approach [16] [17] uses tolerance classes in Rough Set model to generate groups of similar terms. Representation of a document as a document term vector leads to high dimensionality problem as there will be enormous number of index terms in corpus. It is possible to reduce the dimensionality by considering equivalence classes of terms associated with related concepts as a single dimension for Rough Set model.

In this paper the authors propose a hybrid clustering approach to group the terms based on their co-occurrence in various documents. This approach adopts the outcome of hierarchical clustering as seed points to further proceed for partitioned clustering. This provides maximum versatility with respect to the types of objects that can later be compared to each other. To represent queries and documents by groups of terms, a partition enables comparisons between documents, between queries and between documents and queries. (The section 2 presents the Hybrid clustering approach. In section 3 we will introduce the hybrid clustering technique for generating Concepts. The section 4 presents the illustration. The section 5 presents experimental results and it is followed by the conclusions.)

2. Document Pre-Processing

Any document consists of repeated occurrences of keywords in different forms. Not all words are equally significant for representing the semantics of a document. To arrive at a well defined index terms the documents have to undergo a series of transformations [12], [13] like removing common stop words, stemming process, and stripping least useful keywords. Stop words are information-poor connectives (like articles, auxiliary verbs, etc.) found in many languages. Stop words are removed from a document based on a list of stop words specific to that language (ex. and, or, an in English language). It removes a few hundred dimensions from corpus, it is not enough to make a significant difference. However, in certain cases this approach has a risk of mistaking an information-rich term for a stop word.

Word stemming involves removal of word suffixes, leaving only the stems or roots. Removing suffixes by automatic means is an operation which is useful in the field of information retrieval. In a typical information retrieval environment, one has a collection of documents, each described by the words in the document title and possibly by words in the document abstract. Ignoring the issue of word organization, it is computationally easy to represent a document by a vector of words, or terms. In general terms with a common stem posses similar meaning. For example; the word “CONNECT” is found with different stems like:

CONNECT
CONNECTED
CONNECTING
CONNECTION
CONNECTIONS

Frequently, the performance of an information retrieval system will be improved if term groups such as this are transformed into a single term. This is carried out by removing various suffixes -ED, -ING, -ION, IONS to leave the single term CONNECT. In addition, the suffix stripping process will reduce the total number of terms in the information retrieval system, and hence reduce the size and complexity of the corpus in the system. Obviously, this is language-dependent and cannot be generalized to any language. **Porter stemming algorithm** [11] (or ‘Porter stemmer’) is used in the present work for removing the common morphological and inflexional endings from words. Term normalization is carried out to identify the list of keywords present in the document along with their frequencies.

Stripping least useful keywords: The frequency of a keyword in a document indicates the relevance of the document to the concept associated with the keyword. The keywords that are specific to a limited set of documents (with least frequency) as well as the keywords that occur very often in most of the documents (with very high frequencies) are less likely to be significant in finding the relevance of a document. Hence the keywords that appear in less than three documents and those that appear in more than 40% of the corpus are considered as less significant and removed from the list of keywords.

After applying the above transformations, each document is represented in vector space model constituting the frequencies of the identified list of significant keywords referred to as index terms; e.g. ith document is represented as a vector< 18, 4, 0, 9, 0, 0, 0, 2> to indicate that it contains the 1st index term for 18 times, 2nd index term for 4times and 4th index term for 9times and last index term for 2 times.

The observation that multiple index terms are associated with the same concept is used to find the relevance of a document for a context by placing all those index terms into an equivalence class. Subjectivity of the domain experts is the basis for partitioning the index terms into equivalence classes, in most of the existing information retrieval systems [18] [20]. The number of index terms is enormous there is a need to go for automating the process of partitioning the index terms into equivalence classes, each of which to represent various index terms associated with a single concept.

It is observed that most of the index terms associated with a single concept either occur together or do not occur in a document based on whether the document is relevant or not to that concept. For example if the terms t_i and t_j are associated with the same concept (belongs to the same equivalence class) then the frequency of the item t_i in an arbitrary document is expected to be high if the document has t_j with high frequency. Hence the document wise distribution of an index term is useful for identifying the terms constituting an equivalence class and each term is represented as a vector of its frequency in various documents belonging to the selected corpus. For the purpose of partitioning the terms into equivalence classes, we transform the preprocessed data into term-document matrices, each of which limited to the collection of documents belonging to a selected corpus. Clustering techniques [15] are used for identifying the groups of terms based on their similarity estimates.

3. Generation of Concepts using Hybrid Clustering

Hierarchical clusters presented in the form of trees called dendrograms [4] are of great interest for a number of application domains. They provide a view of the data at different levels of abstraction. In addition, there are many times when clusters have sub clusters, and hierarchical structures represent the underlying application domain naturally. Hierarchical clustering solutions have been primarily obtained using agglomerative algorithms in which objects are initially assigned to their own cluster and then pairs of clusters are repeatedly merged until the whole tree is formed. The selection of clusters to be merged to form larger clusters can be carried on by using single linkage, complete linkage, or average linkage algorithms. In the present work complete linkage algorithm is used for agglomerative clustering till the terms are grouped into required number of clusters. The distance between two clusters is defined as the largest distance between any two points, one each from both the clusters. If C_i and C_J are two clusters, the distance between them is defined as

$$D_{CL}(C_i, C_J)= \max_{a\in C_i,\, b\in C_J}\{ d (a, b)\}$$

Partitional clustering algorithm [5] is applied along with the above algorithm, since the document datasets are relatively large. Partitioning the data samples into equivalence classes is carried out using Incremental K-means algorithm [10] while grouping the terms associated with similar concepts into equivalence classes.

The Incremental K-means algorithm requires the number of clusters to be formed, K, and accordingly those many random points are used as seed points to represent the initial clusters. Each of the remaining points will be included into its nearest cluster based on the distance between the centroid of the cluster and the point getting included. As and when a point is included in a cluster the centroid of a cluster will be readjusted to represent the midpoint. This way of partitioning the data points into a set of clusters can be continued in later iterations by taking the centroid of the set of clusters as seed points for the next iteration. This iterative process terminates when a steady state is reached with no difference in the set of clusters in successive iterations.

However the results of the Partitional clusters are highly influenced by the proper selection of initial seed points. The Incremental K-means algorithm may not generate quality clusters if it starts with randomly generated seed points. One approach is to make use of multiple sets of random seed points as the basis for generation of multiple clustering solutions and selecting the best set of clusters based on their quality. The draw back with this approach is Incremental K-means algorithm has to be executed repeatedly on different sets of randomly generated seed points which requires a lot of computational resources. An alternative approach is to select seed points randomly from a sample dataset instead of the whole set. Though the overhead involved in this method is less it is unlikely that the resulting clusters will be of high quality especially while dealing with formation of equivalence classes containing terms associated with similar concepts. In this paper we present a hybrid clustering technique which uses the merits of the agglomerative clustering as well as Partitional clustering to form quality clusters.

The centroids of these clusters are taken as initial seed points to the incremental K-means algorithm [9]. The clusters generated by Incremental K-means algorithm represent the equivalence classes of similar terms.

A term is represented as an m-tuple of its weight (term weight) in various documents numbered from 1 to m. For example i^{th} term is represented as ti=<w_{i1}, w_{i2}…wim> in a corpus containing m documents. We have experimented with three metrics to estimate the term weight in a document and analyzed the results of the hybrid clustering in each case. The first representation is a vector of frequencies of the term in various documents where i^{th} term is represented as ti=<tfi1, tfi2… tfim>. The second method uses the classical tf-idf estimate of the term weight in a document [12] as a component of the term vector. tf-idf estimate can be expressed mathematically as:

$$tf\text{-}idf= tfij * \log(m/dfi)$$

where tfij is the term frequency of the i^{th} term in j^{th} document
m be the total number of documents in the corpus

dfi= number of documents in which the i^{th} term appears

In this expression the first term tf_{ij} indicates the extent to which the term is present in the document, where as the second term log (m/dfi) indicates how specific the term is in the corpus. The draw back of using the tf-idf for this purpose is that it is ignorant of the size of the document. As a consequence the distance measure calculated based on tf-idf estimates are unduly influenced by large sized documents. To overcome this drawback we suggest a third method which uses tp-idf estimate as a component of a term vector. The tp-idf to represent weight of i^{th} term in the j^{th} document is expressed mathematically as

$$ntf\text{-}idf= (tfij/\sum_{i=1}^{n} tfij)*\log(m/dfi)$$

Where n is a number of index terms relevant for the corpus

The first term in this expression i.e. tfij / $\sum_{i=1}^{n}$tfij represents the probability of a term to occur in a document irrespective of the size of the document. It can be observed that the modification makes the tp-idf estimate insensitive to the size of the document and the distance estimates calculated based on the tp-idf gives equal importance to documents of all sizes.

The term-document vector representation in the above three methods is illustrated with a sample example of 5 terms and 5 documents given in the tables below.

The term weight should reflect the prominence of a term in a document and a zero term weight is given to all those terms of the vocabulary that do not occur in a document. Hence, the term-document vector is expected to be sparse. Though cosine similarity metric, which is the dot product of the unit vectors being matched, is generally used for finding similarity between two document-term vectors, it is not found suitable for matching term-document vectors(see Table 4).

Table 1. Term weight estimates in terms of frequencies

Term\ document weight	d_1	d_2	d_3	d_4	d_5
t_1	3.88	2.9	0	0.10	0.29
t_2	3.395	1.97	0.38	1.36	0
t_3	1.94	2.9	0.58	1.16	0
t_4	2.9	4.07	0	0.58	0.1
t_5	1.16	2.13	0.68	0	0.3

Table 2.Term weight estimates in terms of tf-idf measure.

Term\document coverage	d_1	d_2	d_3	d_4	d_5
t_1	0.0283	0.020	0	0.003	0.04
t_2	0.0248	0.0135	0.022	0.041	0
t_3	0.01416	0.020	0.034	0.035	0
t_4	0.021	0.028	0	0.0176	0.0138
t_5	0.0084	0.0142	0.04	0	0.041

Table 3. Term weight estimates in terms of tp-idf measure

Term\ Document frequency	d_1	d_2	d_3	d_4	d_5
t_1	40	30	0	1	3
t_2	35	20	4	14	0
t_3	20	30	6	12	0
t_4	30	42	0	6	1
t_5	12	22	7	0	3

Hence, we use Euclidian distance for estimating the proximity of terms. The distance between two vectors (a&b) is calculated using the Euclidian distance measure expressed as:

$$d(a,b) = \sqrt{\sum_{i=1}^{n} (b_i-a_i)^2}$$

Accordingly the distance between terms t_2&t_3 is given by disf(t_2,t_3)= sqrt((35-20)^2+(20-30)$^{2+}$+(4-6)^2+(14-12)^2+(0-0)^2) =18.248 when the term is represented as vector of frequencies. Similarly using tf-idf estimate the distance between t_2&t_3 is given by distf(t_2,t_3)=sqrt((3.39-1.94)^2+(1.97-2.9)^2+(.38-.58)^2+(1.36-1.16)^2+(0-0)^2)=1.745. Finally using tp-idfestimate the distance between t_2 & t_3 is calculated by disnf(t2,t3) =sqrt((.0248-.01416)^2+(.020-.0135)^2+(.034-.022)^2+(.035-.041)^2+(0-0)^2)=.01786.

It can be observed that when terms are represented as vectors of frequencies and when terms are represented as vector of tf-idf estimates, the larger documents (d_1, d_2) are influencing the distance between the terms unduly. The Table 4 depicts the proximity of the term, t2 with respect to the terms t1 and t3. The first three rows of the table shows the Euclidian distance between the terms when the term-weights are represented by frequency, tf-idf and ntf-idf estimates. The fourth row gives the cosine similarity between the terms. As given in the column three it can be observed that the term t2 was considered as nearer to t1 in all the methods except in the method proposed by the authors namely ntf-idf estimation of term weights. The prominence of the terms t1 and t2 differ drastically in the smaller documents d3, d4, and d5 and this fact is dominated by the difference in the prominence of these terms in the larger documents d1 and d2. In this way the three existing methods, when applied for finding the proximity of the terms of an vocabulary, do not consider all documents equally i.e. all documents do not contribute equally to the proximity estimation. On the other hand it can be observed that the proposed metric has given due importance to all documents irrespective of their sizes and suggested term t3 as nearer to term t2.

Table 4. Proximity between t1, t2&t3

Method		proximity(t1,t2)	Proximity(t2,t3)	nearest term to t2
Euclidean distance	frequency based	17.86	18.248	t1
	tf-idf based	1.719	1.745	t1
	tp-idf	0.0616	0.01786	t3
Cosine similarity		0.94	0.9065	t1

Once the distance is calculated the terms are clustered using agglomerative approach with complete linkage algorithm until required numbers of clusters are formed. The cluster centroids are taken as seed points for the Incremental K-means algorithm to partition the terms into equivalence classes each of which containing terms associated with related concepts. The results obtained by the hybri clustering approach to form equivalence classes with tf-idf and tp-idfterm weight estimates are found to be promising.

4. Experimental Analysis:

The system is developed using java under the eclipse environment and it has been tested using the Ancillary Data set which contain around 1500 abstracts of the thesis submitted in various universities on various subjects like Machine Learning, Image processing and so on, it has become a popular data set for experiments in text applications of machine learning techniques, such as text classification and text clustering. Documents belonging to each of the corpus are processed separately and are referred to as corpus. After preprocessing, the collection of terms found in various documents of a corpus forms the vocabulary. We have processed 250 documents belonging to Machine Learning group and extracted approximately 1800 index terms. We used the Hybrid clustering algorithm to partition the vocabulary into equivalence classes of terms associated with related concepts. In this process we need to find the number of clusters, K, to optimize Sum of squared error (SSE) and in turns cluster quality. As shown in the graphs in figures [2, 3, and 4] the SSE reduces as the number of clusters increases. The optimal number of clusters (K) is given by such a point at which the rate of change in the SSE with respect to number of clusters reaches saturation. Similarly the terms belonging to the other corpus namely neural networks, AI, e.t.c. are also partitioned into equivalence classes as indicated by the clusters resulted by the application of Hybrid clustering algorithm. The Fig1 depicts the formation of clusters based on the Hybrid clustering algorithm with different approaches to proximity estimation. It clearly shows that the proposed tp-idf estimate for term weights forms well distributed clusters for corpus containing varied sized documents. For this data it was observed that the optimal value of K is around 150 which is approximately 8% of the size of the vocabulary.

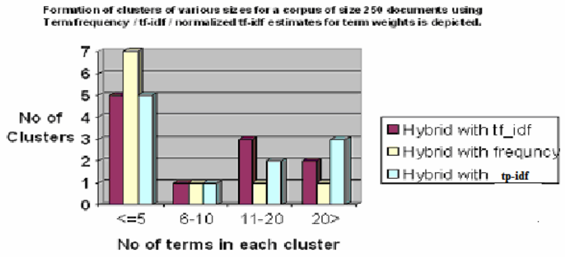

Figure1. Histogram showing the formation of clusters

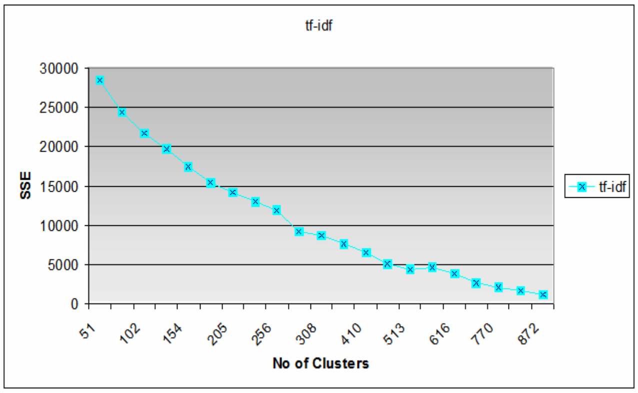

Figure 2: Graph showing SSE for tf-idf term weight

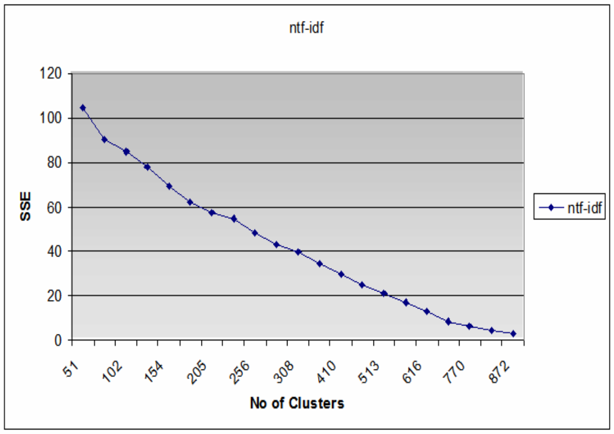

Figure 3: Graph showing SSE for percentage term weight

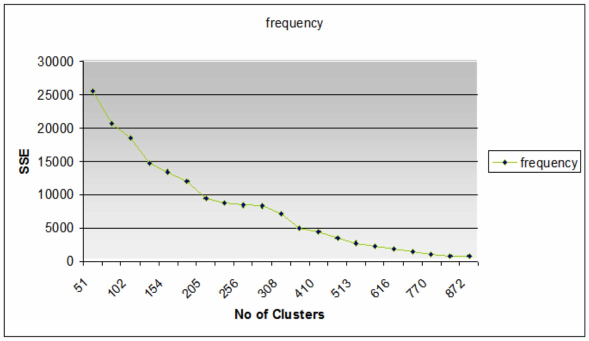

Figure 4: Graph showing SSE for frequency based term weight.

The Rough set based document ranking system is developed using the term clusters resulted from the application of the proposed hybrid clustering algorithm. The performance of the system using term weight estimates for formation of equivalence classes is tested and the implementation of the proposed system with tp-idf is to be the best.

6. Conclusion

We have developed a hybrid clustering algorithm to form equivalence classes of related terms required by a Rough set based document ranking system. A term is represented as a term-document vector containing term weights in various documents. Existing methods of estimating the term weights namely term frequency and tf-idf are found to be ineffective while dealing with documents of varied sizes. We proposed tp-idf estimate for term weights and compared its performance with the existing term-weight estimates. Clustering is performed to group together related terms of a concept into equivalence classes, which can be used to reduce the dimensionality of the documents for rough classification. This improves the performance of Rough set based document ranking system due to the reduced dimensionality.

References

[1] Akiko Aizawa “A Co-evolutionary Framework for Clustering in Information retrieval Systems”, Evolutionary Computation, 2002. CEC ’02. Proceedings of the 2002 congress on Evolutionary Computation, IEEE, Volume 2, 12-17 May 2002 pp. 1787-1792, 2002.

[2] C.J. Van Rijsbergen, “Information Retrieval”, Second edition. London: Butterworths, 1979.

[3] Deerwester, S., Dumais, S., Furnas, G., Landauer, T., Harshman, R. “Indexing by latent semantic analysis”, Journal of the American Society for Information Science, 41(6), pp. 391-407, 1990.

[4] Earl Gose, Richar Johnsonbaugh, Steve Jost,, “Pattern Recognition and Image Analysis”, Prentice-Hall.Inc., (1997).

[5] I.S.Dhillon, Y.Guan, and J.Kogan “Refining clusters in High Dimensional Text Data”, Proc.Workshop Clustering High-Dimensional Data and its Applications, 2nd SIAM Int’l Conf.on Data Mining, 2002, pp. 71-82 2002.

[6] Jiawei Han, Micheline Kamber, “Data Mining Concepts and Techniques”, Morgan kaufmann publishers (2006).

[7] Lech Polkowski, Shusaku T Sumoto, T sau Y.Lin, “Rough Set Methods and Applications –New Developments in Knowledge discovery in Information Systems”, A Springer-Verlag Company (2000).

[8] L.J.Mazlack, Aijing He, Y Zhu, Sarah Coppock, “A Rough Set Approach in Choosing Partitioning Attributes”, Proceedings of the ISCA 13th International Conference (CAINE-2000), November, 2000, pp.1-6, 2000.

[9] Padmini Das Gupta, “Rough sets and information retrieval”, JACM, vol.88, pp. 567-572, 1988.

[10] Pang-Ning Tan, Michael Steinbach, Vipin Kumar, “Introduction to Data Mining”, Addison Wesley publisher, pp. 508 2006.

[11] Porter.M.F. “An algorithm for Suffix Stripping”, Program 14(3), pp.130- 137, 1980.

[12] Ricardo Baeza-Yates, Berthier Ribeiro-Neto, “Modern Information Retrieval”, Pearson Education (2004), pp. 165-173.

[13] Richard K Belew, “Finding Out About-A Cognitive Perspective on Search Engine Technology and the WWW”, Cambridge Press, pp. 44-47, 2000.

[14] S.K.M.Wong, W.Ziarko, “A machine learning approach to information retrieval”, ACM Conference on research and development in information retrieval, 228-233, 1986.

[15] S.M.Ruger,S.E.Gauch , “Feature Reduction for document Clustering and Classification”, Technical Report DTR 2000/8; Department of Computing, Imperial College; London, England, 2000.

[16] Tu Bao Ho,Saori Kawasaki,Ngoc Binh Nguyen, “Non hierarchical Document Clustering Based on Tolerance Rough Set Model”, International Journal of Intelligent Systems, Vol. 17, No.2, 199-212, 2002.

[17] Tu Bao Ho,Saori Kawasaki,Ngoc Binh Nguyen, “Text Mining with Tolerance Rough Set Models”, International Journal of Intelligent Systems,2002.

[18] W.Shang, H.Huang, H.Zhu, “A Novel Feature Selection Algorithm for Text Categorization”, Journal on Expert Systems with Applications, Elsevier Publisher 2006.

[19] Ying Zhao, George Karypis, “Hierarchical Clustering Algorithms for Document Datasets”, Data Mining and Knowledge Discovery, Springer science publishers vol 10, pp.141-168, 2005.

[20] Yun Li, Zhong-Fu Wu, Jia-Min Liu, Yan-Yun Tang, “Efficient Feature Selection for High-Dimensional Data Using Two-Level Filter”, In the Proceedings of the Third International Conference on Machine Learning and Cybernetics, Shanghai, 26-29 August 2004.

K. Thammi Reddy Presently working as an Associate Professor in the Dept of Computer Science and Engineering, College of Engineering, GITAM, Visakhapatnam. A.P, India. He is pursuing his Ph.D from JNTU, Hyderabad. His areas of interest include Data Mining, Information Retrieval, and Data base systems. He is active member in professional bodies like ISTE, IE, IETE, and CSI.

Prof. M.shashi is working in the Dept.of Computer Science & Systems Engineering in AndhraUniversity, Visakhapatnam for the last 21 years. She received AICTE career award as young teacher in 1996. She published technical paper in national & international journals and co-authored Indian Edition of text book on “Data Structures and Program Design in C” from Pearson Education Ltd. Her areas of interest include Data Mining, Artificial Intelligence and Machine Learning. She is active member in professional bodies like ISTE, IE, IETE, and CSI.

Dr. L. Pratap Reddy, received the B.E. degree from Andhra University (INDIA) in Electronics and Communication Engineering in 1985, the M.Tech. degree in Electronic Instrumentation from Regional Engineering College (WARANGAL) in 1988 and the Ph.D. degree from Jawaharlal Nehru Technological University (HYDERABAD) in 2001. From 1988 to 1990 he was lecturer in ECE Department of Bangalore Institute of Technology (BANGALORE), from 1991 to 2005 he was faculty member in JNTU College of Engineering (KAKINADA). Since 2006 he is with Department of Electronics and Communication Engineering at JNTU, Hyderabad. His current activity in research and development includes, apart from telecommunication engineering subjects, Image Processing, Pattern Recognition and Linguistic processing of Telugu language. He published 25 technical papers, articles and reports. He is active member in professional bodies like ISTE, IE, IETE, and CSI.