International Journal of Applied Engineering Research ISSN 0973-4562 Volume 14, Number 11 (2019) pp. 2782-2788
© Research India Publications. http://www.ripublication.com

Word Ranking based Document Clustering

Dr. Ratnesh Prasad Srivastava

Department of Information Technology, GBPUAT Pantnagar, Uttarakhand 263153, India.

Dr. Rajiv Singh

Department of Electrical Engineering, GBPUAT Pantnagar, Uttarakhand 263153, India.

Abstract

The statement, that information is growing has be- come an understatement, considering the developments over the past decade. In fact information is burgeoning. The vast plethora of existing information over the internet and the continuing proliferation of data every day calls for newer ways to handle this information overload. The information overloaded on the internet does not follow any standard of arrangement of text and also suffers with the problem of immensity. There is also a missing mechanism of archiving and organizing information over the internet. Digital Libraries which encompass thousands of books and documents require the services of tools which can identify information across various levels of granularity (as in books, chapters or even pages). Manual data creation for an exponentially increasing volume of digital objects is becoming increasingly difficult. We develop a document clustering engine which uses a word ranking module with document categorization and clustering in order to recognize objects during information overload.

Keywords: Clustering, Categorization, Word Ranking, Pre Pro- cessing, Cluster Node, WoRm Framework.

I. INTRODUCTION

The amount of information available to us has become unmanageable due to the advent of Internet. The need of classification arises to classify the similar type of document based on the complexity of data it contains and the details of the document. Therefore document classification is done to represent two different types of analysis which is related to document categorization and clustering [1]. Categorization [2] and Clustering is a mix of supervised and unsupervised approach used for Document Clustering. It is also required to design a high quality document classification [3] mechanism which may be helpful in finding useful document and information contained in them. We have used an approach of word ranking to develop a document classifier. Document categorization is a process where documents are assigned to predefined set of categories. It can be viewed as a two-step process where a prerequisite is a substantial set of training documents which are already categorized into the predefined sets. Often this set of categories is defined previously and remains unchanged. The first step is to train the Document Categorizer based on various available classifiers using the training documents. The second step, however, categorizes new documents into the sets using the trained classifier. The main drawback of categorization techniques is that the set of categories is pre-defined and could not be modified.

This however is an obvious shortcoming keeping in mind the rapid increase of information on the Internet. As opposed to categorization, Clustering is an unsupervised learning procedure. Clustering does not require any pre-categorized training documents. Clustering is a one step process groups all and only relevant documents into clusters according to their properties. Any cluster analysis method requires some measures to be defined on the objects that have to be clustered and a threshold value indicating the similarity or dissimilarity between them. The objective is that each cluster is a collection of objects that are similar to each other within the same cluster and dissimilar to the members of other clusters. Clustering methods can be divided generally into hierarchical and partitioned clustering methods. Within both types there exist several variants for defining the clusters. This paper is organized as followed: section II discuss the literature surveyed in the field of categorization and clustering, section III architecture related to Document Clustering Engine(DocClus), section IV discuss the modules of DocClus and finally section V discuss the result and discussion and section VI finally discusses the conclusion and future work.

II. LITERATURE SURVEY OF CATEGORIZATION AND CLUSTERING

Document Categorization is a mechanism where document is categorized to be assigned with a set of predefined categories. The first step is to train the Document Categorizer based on various available classifiers using the training documents. The second step, however, categorizes new documents into the sets using the trained classifier. Many approaches have been put forward for Categorization, the most popular of which are, Decision Trees Gerst [2], Bayesian Approach Domigos [3], Neural Networks Goh, et al [4], Regression based Methods Yang et, al [5], Vector-based Methods (Centroid based and SVM's Joachims [6]).

In [7] Brucher et. al, gave a brief overview of classification. Both Clustering (unsupervised) and categorization

[1] Ratnesh Srivastava is with the Department of Information Technology, College of Technology, G.B.Pant University of Agri. & Technology, Pantnagar, 263145, Uttarakhand, India, e-mail: write2ratnesh@gmail.com.

[2] Dr. Rajiv Singh is with the Department of Electrical Engineering, College of Technology, G.B.Pant University of Agri. & Technology, Pantnagar, 263145, Uttarakhand, India , e-mail: rajiv77singh@gmail.com.

(supervised) approaches are elaborately explained along with a comparison of the same. They reported SVM based methods clearly outperform the other categorization methods. The main drawback of categorization techniques is that the set of cate gories is predefined and could not be modified. This however is an obvious shortcoming keeping in mind the rapid increase of information on the Internet.

Clustering does not require any pre-categorized training documents. Clustering is a one step process groups all and only relevant documents into clusters according to their properties. Any cluster analysis method requires some measures to be defined on the objects that have to be clustered and a threshold value indicating the similarity or dissimilarity between them.

The objective is that each cluster is a collection of objects that are similar to each other within the same cluster and dissimilar to the members of other clusters. Clustering methods can be divided generally into hierarchical and partitioned clustering methods. Hierarchical Clustering methods use distance or similarity matrix of documents. The number of clusters can be selected as required after the clustering is done. Two basic types of hierarchical clustering are - Agglomerative and Divisive approaches. Agglomerative clustering uses a bottom-up approach and either of single, complete or average link approaches are used to combine the clusters. Divisive clustering uses a top-down approach where initially all the documents are grouped into a cluster and gradually the clusters are broken into smaller ones which are more accurate. The main advantage of hierarchical clustering is the low computation power requirement which gives it a distinct advantage on the Internet which puts a limitation on the computational power. This also has some disadvantages. Each step is decisive and not reversible. The technique is not robust and it has no protection against outliers.

Partitioned clustering uses features [4] from each document to construct a feature vector matrix. So, each document is represented as a vector in the feature space. Clustering of the document proceeds by comparing these feature vectors of the documents. The number of clusters to be formed must be given as a parameter beforehand. Various algorithms for partitioned clustering are studied like k-means, single-pass, nearest neighbor, etc. K-means is an iterative procedure of clustering. Hierarchical Document Clustering is not efficient while handling high dimensionality, and high volume of data. In order to improve ease of browsing, and add meaning to clus- ter labels Fung, et al. [1] elaborates on frequent item set-based Hierarchical Clustering (FIHC) methods. Frequent Item set is a set of words which occur with a minimum threshold frequency in all the documents of the cluster. FIHC assigns documents to the best cluster from among all available clusters (frequent item sets). The main idea is that some frequent item sets exist for each cluster in the document set, but different clusters definitely share quite a few frequent item sets. FIHC uses frequent item sets to generate clusters and to organize clusters into a topic hierarchy. FIHC uses only the most commonly occurring frequent items as document vectors instead of the whole lexicon, thereby reducing the dimensionality of the vector space.

III. ARCHITECTURE OF DOCUMENT CLUSTERING ENGINE

The architecture of the DocClus (Document Clustering Engine) is depicted in the diagrams given below. The Document Clustering Engine is an independent entity, the main aim of which is to take a set of raw documents as input and partition them into clusters so that each cluster contains documents which are related to each other in some way.

First the documents are passed onto a preprocessing module which does some preprocessing and each raw text is converted into an intermediate format which is then passed on to the WoRM framework. A detailed list of words along with their weighted scores is obtained as a result. Then the engine a modified hierarchical clustering algorithm to form a hierarchical cluster tree as an end result. The class diagram shown in Figure 1 is a fairly and simple and straightforward structure. The clustering process is done by the Clustering class which uses utility classes such as Cluster Node and Cluster Tree.

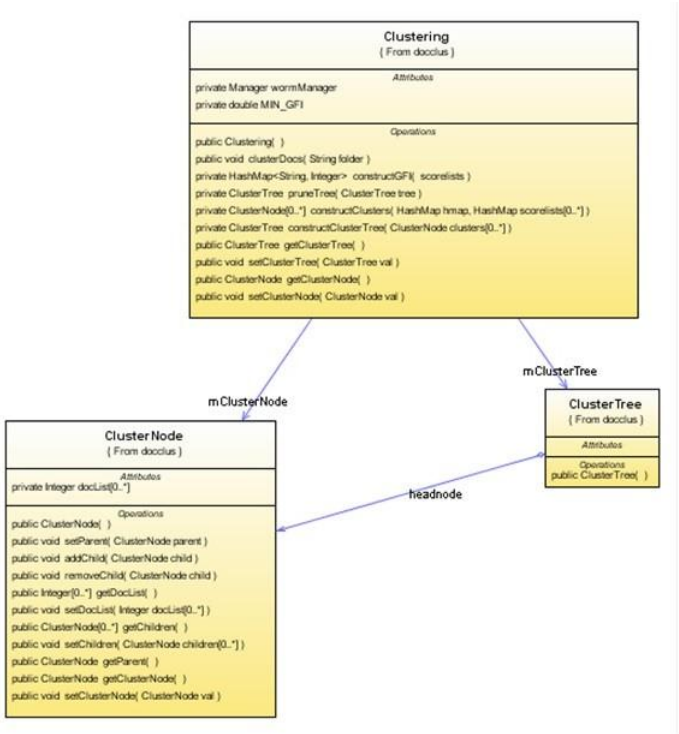

Fig. 1: Class Diagram of DocClus Engine

IV. MODULES OF DOCCLUS

A. Pre Processing Module

The JAVARAP API was integrated successfully with the Stanford NER engine on linux platform. The JAVARAP API is used to do the following tasks: a. Sentence splitting b. Anaphora resolution The Stanford NER API classifies the named entities as PERSON, LOCATION or ORGANIZATION. This information is added to the anaphora resolved input. A Charniak parser is used to help JAVARAP in the anaphora resolution. The API helps give us the set of resolved anaphora pairs after taking the parsed and POS tagged input from the Charniak parser. Manager classes were built to

handle the interface between the APIs. It handles the input files as well as releases the final output file which serves as input to the next word-ranking module. Annotation and cleaning of Data is done by the preprocessing module. The figure 2 works by accepting Input to this module would a raw text document and output will be a document annotated with sentence markers, Parts – of - Speech tags, Named Entity tags and anaphora resolution. Pre Processing module contains various subcomponents, through which the document is passed sequentially leading to the final output.

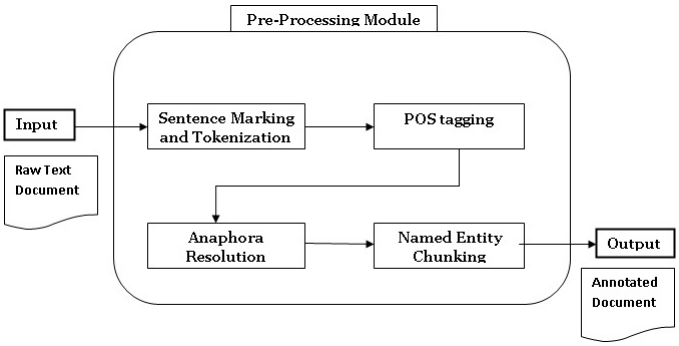

Fig. 2: Pre Processing Module Architectural View

B. Word Ranking Module

The main aspect of Document Summarization and Document Clustering has always been extracting the most salient words from the given text. The performance of both of them depends heavily on the quality and information represented by the words being extracted [10]. Many approaches have been followed, especially in document summarization, varying from simple word frequency based techniques to methods based on semantic importance of words and their occurrences like Lexical Chains and Cue Phrase Methods. Words have different levels of meaning and importance associated with them. Some of these meanings although not exhaustive are briefly presented below.

Frequency – Words occurring more frequently than others tend to be more important.

- Positional importance- words occurring as parts of the title or section headings, the first few sentences of a paragraph, etc are generally more salient than the others.
- Lexical importance- Nouns and verbs are attach themselves with more importance than adjectives, adverbs and other lexical categories [8][9]
- Syntactic and Semantic Importance- The head nouns and main verbs of a sentence are more salient than the other nouns and verbs.
- Discourse level importance- Words occurring in a particular context or more precisely, occurring near other topically salient words/phrases should be more important than others.

Most of the work done in word extraction based summarization systems has focused only on a handful of these features. Our work tries to explore the effect of all of these features on the production of summary. We intend to include five features which would cover all the cases discussed above and rank the words in the document independently. Then a cumulative score is obtained from a linear combination of all the scores. The main task of this module is to pick out the most salient words from the annotated document and rank them.

Initially, Stop Word removal helps in removing irrelevant words. The remaining tokens are then subjected to a stemming procedure in order to reduce to its morphological root. Then ranking is done based on a linear combination of five different features which take into account various levels of importance of a word. The flow of data through the module is depicted in the figure 3.

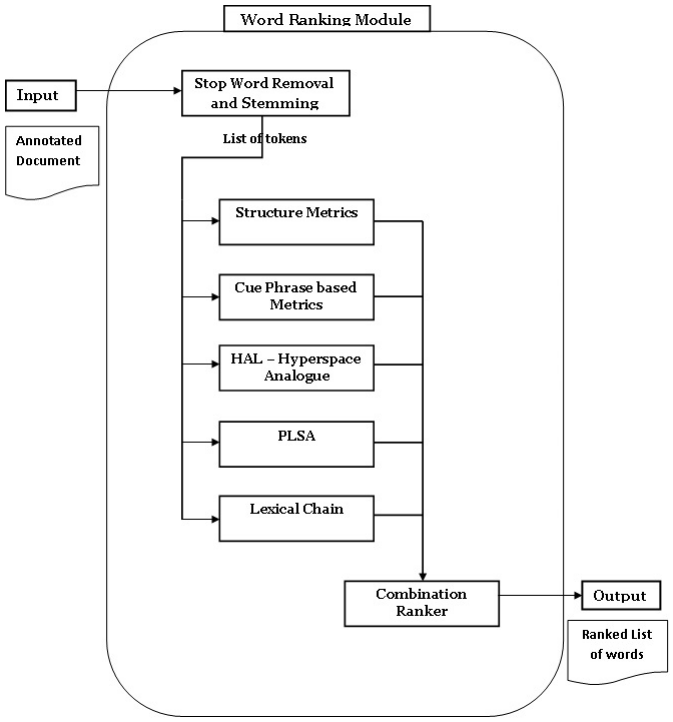

Fig. 3: Word Ranking Module (WoRM) Architectural View

The candidate words are the nouns in the input documents (which were earlier POS tagged in the Pre-processing stage). Additionally, we plan to include noun compounds like "quantum computer". Many noun compounds and collocations do not appear in WordNet. This issue would be addressed by using a shallow parser. The compound "digital computer" taken as a single candidate word and not as two different words reducing the number of candidate nouns. Compounds are given the sense of their head noun. In this case, it is 'computer'. Clearly, this is a better choice than giving the sense of 'digital'.

The implementation of the Word Ranking Module is done in a very modular architecture. A central Manager class is

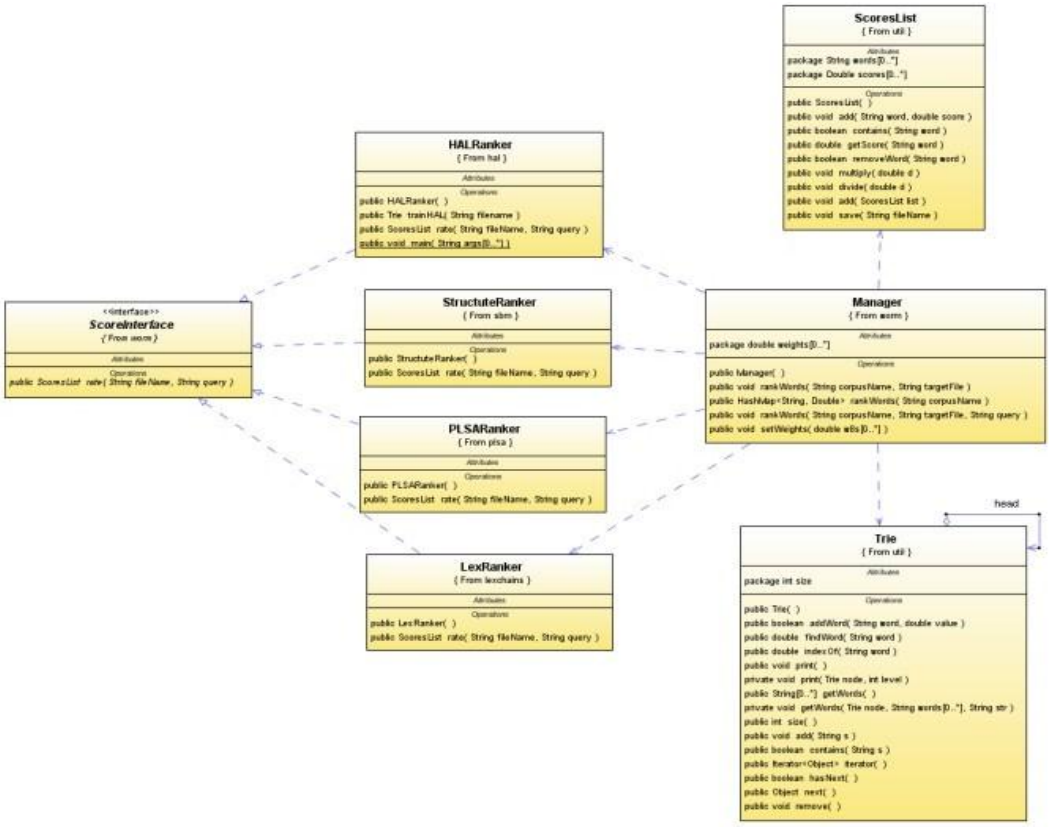

Fig. 4: Class Diagram of Word Ranking Module (WoRM)

responsible for regulating control throughout the module. The WoRM module is based on several features which give weighted scores to words in a document. Provision for adding new features is handled in an elegant way using java interface Score Interface. Any implementation of a particular feature should realize this interface. The Manager deals with the Score Interface instances rather than the actual feature instances.

This way adding a new feature would not affect the functioning of the manager. A detailed class diagram is depicted in figure 4 below. The Manager class and the other feature classes are assisted several utility classes. The main purpose of this Module is to extract the most important words from a document and give weight and scores to each word. This is done by the method "rank Words" in the Manager class. In the word ranking module, the weights of each Ranking algorithm (PLSA, Lexical Chains, etc.) can be further customized with further testing.

C. Document Clustering Module

We propose to build a Document Classifier which would group multiple documents into several clusters such that each cluster contains documents which are relevant to themselves but not to documents from other clusters. The basic architecture of document classifier is depicted in figure 5 below. The input is given as a set of multiple text documents. Every single document is processed initially by the pre-processing module. It would result in annotation of the documents as described earlier. These annotated documents are then passed on to the word ranking module which would give a list of words forming the lexicon of the text along with their ranks. The ranking of these words are calculated by forming a linear combination of all the scores obtained by each feature.

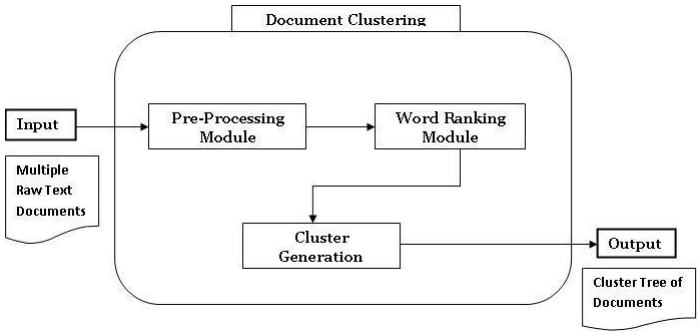

Fig. 5: Document Clustering Architectural View

Every document now has its own set of ranked words which represent it. In order to cluster these documents we follow a hierarchical clustering approach which is a slight modification to Frequent Itemsets Hierarchical Clustering (FIHC) as suggested by Fung, et all []. They propose that the accuracy of the Clustering can be improved by first discovering the hidden topics among the documents and then constructing clusters based on them. FIHC actually assigns a document to a best cluster from all the available clusters. The modification we propose for FIHC is to select the ranked list of words to

represent the clusters instead of the frequent item sets. This helps in a better representation of the documents as we take into account several other factors in selecting the words rather than just frequency as in FIHC. We propose to call this method Ranked Item set Hierarchical Clustering. The rest of the algorithm used is more or less similar to FIHC described by Fung et al [3]. Before we proceed with the algorithm a few terms need to be defined.

(a) Global ranked item set – a set of words which appear in the ranked list of more than a minimum threshold of the entire document set.

(b) Global k-ranked item set – the set of top k ranked items in a global ranked item set.

V. RESULT & DISCUSSION

A detailed summary in tabular form of experimental setup design is presented in figure 6.

Comparison done for?	Comparison made on the basis of									
	I	II		III		IV	V			
Threads Response Time	No. of Threads	Time In Running Threads		Average Time In Running Threads		% Saving in Time	Response Time reduction factor			
	100	C	NC	C	NC	From C to NC	From C to NC			
		Figure 6		Figure 7		Figure 8	Figure 9			
	The comparison presented in graphs using figures 6, 7, 8 and 9 presents the need of using asynchronous communication in PBAF.									
Service Switching Time on documents during categorization	Comparison made on the basis of									
	I	II	III	IV		V				
	No. of robots	No. of Services	Service Types	Switching Services based on cores		Average Time in Switching Services Asynchronously				
Figure 10, 11, 12	100	4	Clustering, Categorization	Two	Three	Clustering	Categorization			
Notation Used in this table										
C for Clustering			NC for Non Clustering							

Fig. 6: Summary of experimental setup design.

Before comparing the results obtained by using Clustering Engine a comparison of non clustering and clustering implementation using the multi thread based approaches was necessary on the basis of response time taken by threads implemented using non clustering and clustering based communication. This comparison would set the bench mark to compare the developed Clustering Engine which was designed with the aim to overcome the extreme limitation of callback hell inherited by conventional clustering threads and at the same time, is also comparable with it on the criterion of response time. The experiment was run for 5, 10, 20, 30, 50 and 100 threads for both the cases independently. As shown in the figure 7, the non clustering case takes significantly more response time in comparison to clustering case as expected.

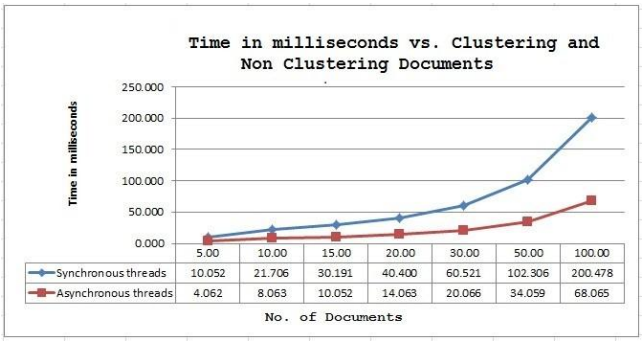

Fig. 7: Comparison between non clustering thread and clustering thread.

Moreover the average response time per thread in case of clustering case is significantly maintained low in comparison to response time in non clustering case as shown in figure 8.

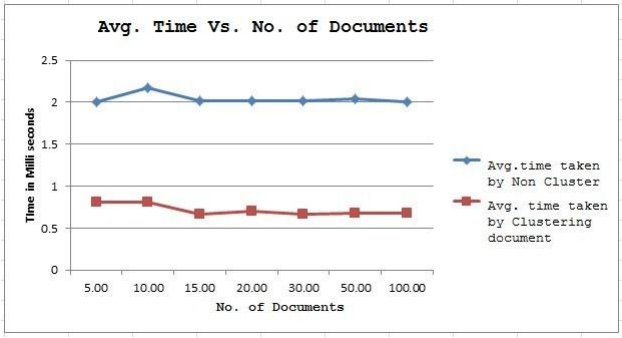

Fig. 8: Average response time by non clustering and clustering threads.

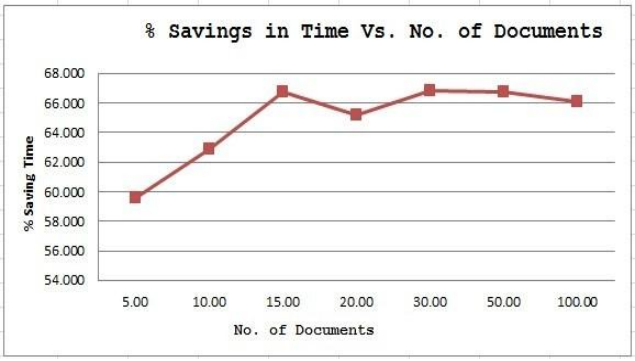

Fig. 9: Percentage saving in time by using clustering over non clustering threads.

This is evident from figure 9, where the % saving in response time for different number of threads is shown and it is found that the saving is significant ranging from minimum 59.59% to maximum 66.8%. In all the cases with different number of threads response time is reduced by a factor of approximately 2.5 to 3 in clustering case as shown in figure 10.

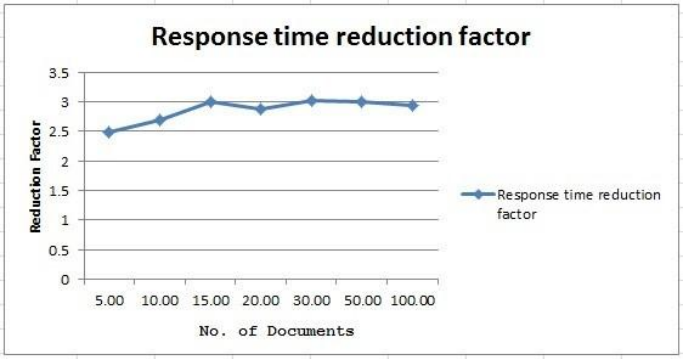

Fig. 10: Response Time Reduction Factor in A non clustering/Clustering thread

Figure 11, compares the average time taken by the 100 documents, in switching of the services by using the Clustering and Categorization based implementation. The average service switching time per document in case of clustering is significantly maintained low in comparison to service switching time in categorization case, shown in the same figure.

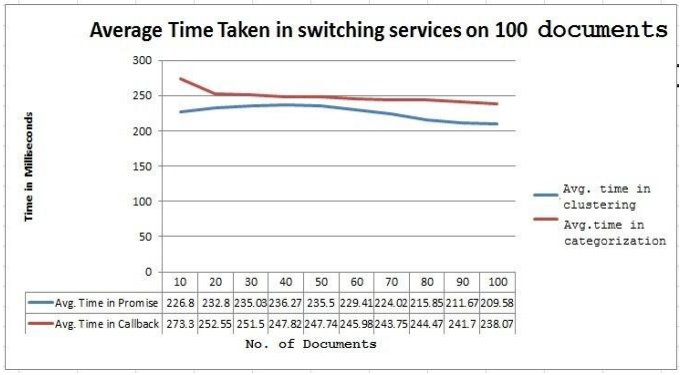

Fig. 11: Average Time taken, during Clustering and categorization, by 100 documents.

In all the cases with different number of documents the service switching time during categorization is reduced by a factor of approximately 1.048 to 1.205 in promise case as shown in figure 12.

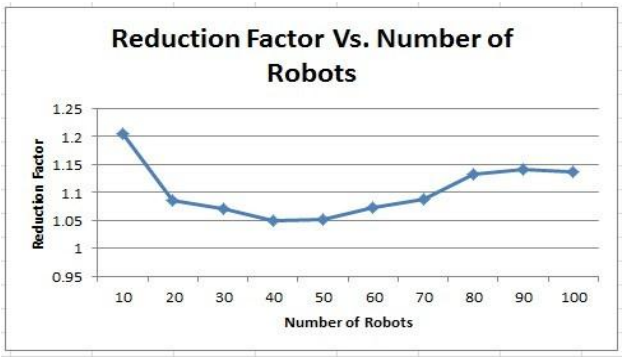

Fig. 12: Switching Time Reduction Factor in Clustering to Categorization based service switching.

VI. CONCLUSION & FUTURE WORK

This shows that clustering service has significant advantage in terms of service switching time over categorization based services. Therefore the developed software Clustering Engine, maintains the response time, completion time and service switching time effectiveness better than conventional implementation. Further the work would include finding a suitable algorithm to verify grammatical correctness of constructs like clauses while trimming and reducing sentences. The Charniak Parser is relatively slow at parsing the input. It needs to be verified empirically as to what advantages other parsers would present if integrated in place of Charniak.

REFERENCES

[1] Gerstl, P., Hertweck, M., Kuhn, B., Hierarchical document clustering", the Encyclopedia of Data Warehousing and Mining, Idea Group Reference, 2004.

[2] Fung, K. Wang and M. Ester, Text Mining: Grundlagen, Verfahren und Anwendungen", in: Praxis der Wirtschaftsinformatik- Business Intelligence, Vol. 39, No. 222, pp. 38-48., 2001.

[3] Fung, K. Wang and M. Ester, "On the Optimality of the Simple Bayesian Classifier under Zero-One Loss", in: Machine Learning, Vol. 29, No. 2-3, pp. 103-130, 1997.

[4] Ng, H. T., Goh, W. B., Low, K. L., " Feature Selection, Perceptron Learning, and a Usability Case Study for Text Categorization", in: Proceedings of the 20th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, pp. 67-73, 1997.

[5] Yang, Y., Chute, C., An Example-Based Mapping Method for Text Categorization and Retrieval", in: ACM Transactions on Information Systems, Vol. 12, No. 3, pp. 253-277, 1994.

[6] Joachims, T., Text Categorization with Support Vector Machines: Learning with Many Relevant Features", in: Proceedings of the 10th European Conference on Machine Learning, pp. 137-142, 1998.

[7] Brucher, et all Heide Brucher , Gerhard Knolmayer and Marc-Andre Mittermayer, Document Classification Methods for Organizing Explicit Knowledge", Proceed- ings of the Third European Conference on Organizational Knowledge, Learning, and Capabilities, 2002.

[8] Barzilay, Regina and Michael Elhadad, Using lexical chains for text summarization", In Proceedings of the Workshop on Intelligent Scalable Text Summarization, pages 10-17, Madrid, Spain, August. Association for Computational Linguistics. 1997.

[9] M. Brunn, Y. Chali, C.J. Pinchak., Text Summarization Using Lexical Chains.", In Workshop on Text Summarization, ACM SIGIR Conference New Orleans, Louisiana USA., September 13-14, 2001.

[10] Bruce Croft B Fei Song., A general language model for information retrieval", In Proceedings of the eighth international conference on Information and knowledge management, pages 316-321, 1999.

AUTHORS' BIOGRAPHIES

Dr. Ratnesh Srivastava is currently a research scholar at Indian Institute of Information Technology, Allahabad and full-time Assistant Professor in the depart- ment of Information Technology, College of Technology, GBPUAT, Pantnagar since 2011. Prior his current job , he was working with software industry for 08 years.

He has experience of using Java based technologies on various domains including health and banking.

Dr. Rajiv Singh, is presently with the Department of Electrical Engineering, College of Technology, G.B. Pant University of Agriculture and Technology, Pantnagar, Uttarakhand. He has around twelve years teaching and research experience. The areas of his research interests include Control, Instruments and Sensors, Wind, PV and other renewable energy systems, Energy policy studies, Smart materials etc. He has several publications in peer-reviewed journals/conferences of national and international repute along with book chapters published by reputed international publishers. He has also attended several national and international conferences in India.