ENHANCED BONDING BASED WEB PAGE INFORMATION RETRIEVAL USING CLUSTERING ALGORITHM

D. Akila*

Dr. C. Jayakumar**

ABSTRACT

In the rapid development of internet technologies, search engines play a vital role in information retrieval. To provide efficient search engine to the user, Enhanced Bond Based Search Engine (EBBSE) for information retrieval has been developed. The traditional search engines provide users with a set of non-classified web pages to their request based on its ranking mechanism. In order to satisfy the needs of the user, an enhanced search engine EBBSE has been proposed. The improvement of information retrieval process can be divided into two parts such as: Extraction &comparison of co-occurrence terms and clustering of documents. In this information retrieval, the relevancy of documents is obtained based on the number of occurrences of each co-occurrence term (in- bonds and out -bonds) in a particular web page and also the text in the web pages. A Tree is generated based upon the threshold value of each and every document. After that, a boost up factor is given to a web page based on the relevancy of content from title and summary. Here the Spam Like pages is clustered as Bad Page Set (BPS) to avoid the inconvenience. The documents can be classified into most relevant, relevant and irrelevant clusters. K- Means clustering algorithm is used to cluster the relevant web pages in order to increase the relevance rate of search results and reduce the computational time of the user.

***Keywords**: Clustering, Information retrieval, Information Extraction, K-Means algorithm.*

*(Research Scholar, Bharathiar University, Coimbatore, Tamilnadu), Assistant Professor, Department of Computer Applications & IT, Guru Shree Shanti Vijai Jain College for Women, Chennai, Tamilnadu.

**(Research Supervisor, Bharathiar University, Coimbatore, Tamilnadu), Professor, Department of Computer Science and Engineering, R.M.K Engineering College, Kavarai Pettai, Tamilnadu.

I. INTRODUCTION

The World Wide Web is a vast resource of information and services that continues to grow rapidly. Powerful search engines have been developed to aid in locating unfamiliar documents by category, contents, or subjects. However, queries often return inconsistent results, with document referrals that meet the search criteria but are of no interest to the user [5]. While it may be currently feasible to extract in full the meaning of an HTML document, intelligent software agents have been developed which extract features from the words or structures of an HTML document and employ them to classify and categorize the documents [5]. Under classification, the researcher attempts to assign a data item to a predefined category based on a method that is created from pre-classified training data (supervised learning). Clustering's goal is to separate a given group of data items (the data set) into groups called clusters such that items in the same cluster are similar to each other and dissimilar to items in other clusters or to identify distinct groups in a dataset [3]. The results of clustering could then be used to automatically formulate queries and search for other similar documents on the web, or to organize bookmark files, or to construct a user profile. In contrast to the highly structured tabular data upon which most machine learning methods are expected to operate, web and text documents are semi structured. Web documents have well defined structures such as letters, words, sentences, paragraphs, sections, punctuation marks, HTML tags and so forth. Hence, developing improved methods of performing machine learning techniques in this vast amount of non-tabular, semi structured web data is highly desirable. In this work solutions to problems such as high dimensionality and scalability associated with existing techniques of mining web documents on the web were provided by proposing an improved data clustering algorithm.

II. RELATED WORKS

Document clustering is widely applicable in areas such as search engines, web mining, information retrieval and topological analysis. Most document clustering methods perform several pre-processing steps including stop words removal and stemming on the document set [4]. Each document is represented by a vector of frequencies of remaining terms within the document. Some document clustering algorithms employ an extra pre-processing step that divides the actual term frequency by the overall frequency of the term in the entire document set. The problem of clustering has been addressed in many contexts and by researchers in many disciplines; this reflects its broad appeal and usefulness as one of the steps in exploratory data analysis. It has great potentials in applications like object recognition, image

segmentation and information filtering and retrieval [1].Most of the clustering techniques fall into two major categories, and these are the hierarchical clustering and the partition clustering [1]. Hierarchical clustering can further be divided into agglomerative and divisive, depending on the direction of building the hierarchy. Hierarchical techniques produce a nested sequence of partitions, with a single all inclusive cluster at the top and singleton clusters of individual objects at the bottom. These algorithms start with the set of objects as individual clusters, then, at each step merges the two most similar clusters. This process is repeated until a minimal number of clusters have been reached, or if a complete hierarchy is required then the process continues until only one cluster is left. These algorithms are slow when applied to large document collections; single link and group-average can be implemented in O+ (n2) time (where n is the number of items) [7], while complete link requires O (n3) time and therefore tends to be too slow to meet the speed requirements when clustering several items. In terms of quality, complete links tend to produce "tight" clusters, in which all documents are similar to one another, while single link have the tendency to create elongated clusters which is a disadvantage in noisy domains (such as the web), because it results in one or two large clusters, and many extremely small ones[7]. This method is simple but needs to specify how to compute the distance between two clusters. The three commonly used methods for computing distance are the single linkage, complete linkage and the average linkage method respectively. Divisive hierarchical clustering methods work from top to bottom, starting with the whole data set as one cluster, and at each step split a cluster until only singleton clusters of individual objects remain. They basically differ in two things, (I) Which cluster to split next (ii) How to perform the split. A divisive method begins with all patterns in a single cluster and performs the split until a stopping criterion is met [7]. Partition clustering algorithms work by identifying potential clusters while updating the clusters iteratively, guided by the minimization of some objective function. The- most common class are the K-means and its variants, K-means, according to [10], is a linear time clustering algorithm. It is a representative of the partition based algorithms where the number of clusters needs to be fixed in advance, it uses a minimum "within class sum of squares from the centres" criterion to select the clusters. K-means, according to [15], is a partition algorithm that derives clusters based upon longest distance calculations of the elements in a dataset, and then it assigns each element to the closest centroid (the data points; that is the mean of the value in each dimension of asset of multidimensional data points). However, according to [16], this method has since been refined and can deal with ellipse shaped data clusters as well as ball shaped ones and does not suffer from the dead unit problem plague of the earlier K-means algorithm.

Also, this new K-means algorithm performs proper clustering without pre-determining the exact cluster number and it is proven to be efficient and accurate[16].

Experimental results of K-means algorithm have been shown in [15]. In detail, it randomly selects K of the instances to represent the clusters based on the selected attributes; all remaining instances are assigned to their cluster center. K-means then computes the new centres by taking the mean of all data points belonging to the same cluster. The operation is iterated until there is no change in the gravity centres. If K cannot be known ahead of time, various values of K can be evaluated until the most suitable one is found. The effectiveness of this method, as well as of others, relies heavily in the objective function used in measuring the distance between instances.

Several variants of K-mean algorithm have been reported in the literature, such as the K-median. The K-mode algorithm is a recent partitioning algorithm that uses the simple matching coefficient measure to deal with categorical attributes [17]. The K-prototype algorithm by integrated the K-means and the K-modes algorithm to allow for clustering instance described by mixed attributes. Some of them attempt to select a good initial partition so that the algorithm is more likely to find the global minimum value. Another variation is to permit splitting and merging of the resulting clusters, i.e. a cluster is split when its variance is above a specified threshold, and the two clusters are merged when the distance between their centroids is below another pre-specified threshold [17].

Using this variant, it is possible to obtain the optimal partition starting from any arbitrary initial partition, provided proper threshold values are specified. Another variation of the K-means algorithm involves selecting a different criterion function altogether [17]. The dynamic clustering algorithm (which permits representation other than the centroids for each was proposed in [6] and [9] and describes a dynamic clustering approach obtained by formulating the clustering problem in the framework of maximum likelihood estimation [1]. It is less sensitive to outliers than traditional K-means due to the characteristics of the norm. Suffix Tree Clustering **(**STC) is a linear time clustering algorithm that is based on identifying the phrases that are common to groups of documents. A phrase in this context is an ordered sequence of one or more words and a base cluster to be a set of documents that share a common phrase. Suffix tree, as defined by [12], is a concept representation of a tie (retrieval) corresponding to the suffixes of a given string where all the nodes with one ‘child’ are merged with their ‘parents’. It is a divisive method which begins with the dataset as a whole and divides it into progressively smaller clusters, each composed of a node with suffixes branching like leaves as introduced by [10].Buckshot algorithm is an hybrid clustering

method that combines the partitioning and hierarchical clustering methods. More precisely, it is a K-means algorithm where the initial cluster centroids are created by applying agglomerative hierarchical clustering (AHC) to a sample of the document collection [10]. Single pass algorithm attempts to find spherical clusters of equal size [7]. It is an incremental algorithm that uses a greedy agglomerative clustering algorithm, assigning each document to a cluster only once. The first processed document is used to start the first cluster. Every additional document is compared to all existing clusters and the most similar cluster is found. If its similarity to this cluster is above a predefined threshold, the document will be added to that cluster; otherwise it will be used to create a new cluster. Fractionation Clustering Algorithm is an approximation of the AHC where the search for the two closest clusters is not performed globally, but locally, and in bound regions [10]. Fractionation algorithm finds centres by initially breaking the corpus of documents into a set number of buckets of predetermined size [15]. The cluster subroutine is then applied to each bucket individually, breaking the contents of a bucket into yet smaller groups within the bucket. This process is repeated until a set number of groups are found, and this end up as K centres.

III. EXISTING SYSTEM

In general, search engine gets query from the user and it makes a process in order to return results to them. The results may be displayed based on the relevancy of keyword and the content of a web page. Traditional search engines present to the users a set of non-classified web pages based on its ranking mechanism. These set of results may not satisfy the needs of the users. Intelligent Cluster Search Engine (ICSE) provides to the user a set of taxonomic web pages in response to a user's query and it would help the users to filter out irrelevant pages or redundant information. This system filters out the irrelevant information using: knowledge base and use of fast clustering algorithm [3] [8]. In this system, user's query is given to the meta-search engine. Then the clustered document set is created based on the given knowledge base and the clustering algorithm of ICSE.

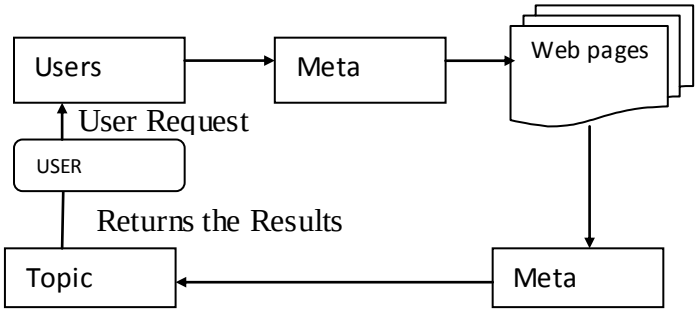

Fig. 1 Design of ICSE

CA-ICSE algorithm is used to cluster the web pages, which increases the relevancy of search results and reduces the computation time [3].This algorithm can be executed in two steps such as: compute the similarity and cluster the pages based on similarity [7]. ICSE system consists of four modules such as: meta-search engine, meta-directory tree, web pages clustering, topic generation.

A. Meta- search engine:

This module uses information extraction technology to parse the web pages and analyze the HTML tags. Stemmer is used to discard the common morphological and inflectional endings and Stop word to discard worthless words, and then the web pages will be converted to a unified format.

B. Meta- directory tree:

In order to cluster the returned web pages rapidly, propose a novel clustering algorithm which uses meta-directory tree as the knowledge base for reducing the computation time required for clustering and enhancing the quality of clustering results.

C. Web pages clustering:

Traditional clustering and classification technologies classify data without a knowledge base. It takes a lot of computation time to find classified results. To avoid this problem, it uses directory-tree approach which can not only cluster the web pages [7] quickly but also assign a meaningful label to each group of classified results.

D. Topic generation:

This module assumes that the words in the web page at the beginning and at the end parts are more important than in the middle part.

IV. PROPOSED MODEL

In the proposed system, K-Means clustering algorithm [13] is used for information retrieval instead of CA-ICSE algorithm. K-Means clustering is more efficient in order to improve the relevancy rate of search results and also in saving computation time. The relevancy rate using CA-ICSE is decreased due to the similarity check between the documents using TF-IDF depending only on the contents. I.e. only the number of occurrences of a given word is compared in each document. So, in some documents the given word may have very low occurrence frequency and in other documents the word may have very high occurrence frequency [9]. Based on ranking the documents are displayed in sequence which may have less similarity documents with highest priority and more similar documents may have least priority. The least similar documents with high priority may lead to dissatisfaction of the

user's needs. So the relevancy rate of documents must be increased in order to satisfy the needs of the users. The efficient way to improve the relevancy rate involves the use of K-Means Clustering algorithm. In the proposed system by using K- Mean's clustering [14] algorithm, the documents are grouped based on the threshold assigned to cluster. Depending on the threshold value of each cluster the documents are selected and the weight of a particular document is compared with the neighbouring documents. The document which gets matched most to that threshold is assigned to the cluster and other documents are discarded. The number of relevant in links and out links of a document is also considered for clustering similar documents. After these clustering, when a user requests for a query only the particular cluster that matches the request is displayed to the user which increases the relevancy rate and reduces the processing time. It improves user satisfaction and provides effective use of search engine.

With the help of this K-means clustering algorithm, the efficiency of search results can be increased. This algorithm performs as follows: when a user provides query to the search engine, it first retrieves all the relevant documents for a query. Meanwhile it retrieves set of co-occurrence terms of the query and it makes comparison of each co-occurring term in the document. When the count of co-occurring terms in the document is obtained, weight of a document can be measured. Then it forms a meta-directory tree with the obtained search results which can be used as a knowledge base for clustering. With the help of this meta-directory tree, the

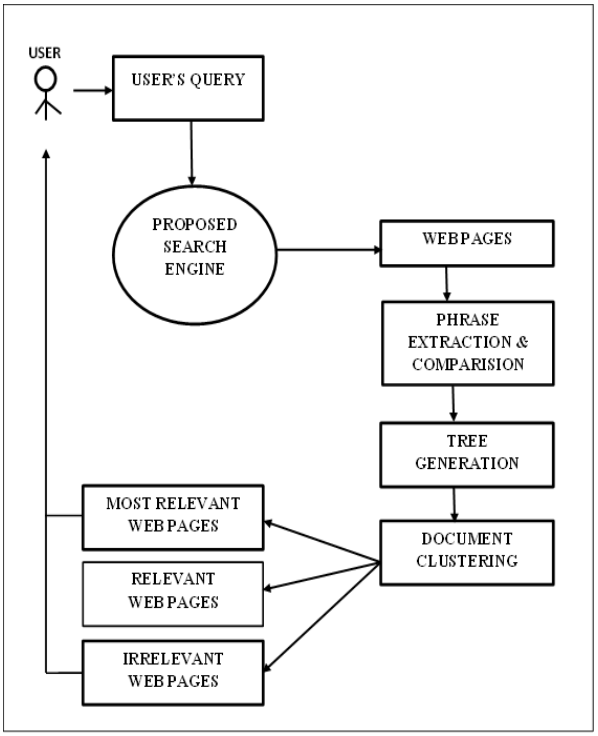

Fig 2: Proposed System

Nodes are compared for relevancy based on the number of in links and out links in a web page. If a node has more number of relevant links then it is treated as a centroid of the cluster. This centroid value is determined by considering the weight of a document. Once the weight is calculated, threshold value for cluster1, cluster2 and cluster3 is assigned as 80, 30 and the values less than 30 respectively which represent the documents with most relevant, relevant and irrelevant clusters. The document which has weight with the centroid is assigned to the cluster and those doesn't support are discarded away from the cluster. The process is repeated until all the obtained results are clustered. Here the spam like pages are eliminated as it increased the computation time.

V. CONCLUSION

In this paper, an approach for efficient retrieval of clustered search results has been proposed, in which the similarity between the documents can be compared by considering the co-occurrence term of a query and not just by the contents of a document. Also the spam pages are eliminated through which the computation time will be reduced. All the documents are compared and the resultant clusters are formed by using K-Means clustering algorithm which improves the relevancy rate and processing time.

REFERENCES

1. B. F. Momin, P. J. Kulkarni, Amol Chaudhari, *"Web Document Clustering Using Document Index Graph"*, IEEE 2006. (Conference Name)

2. Chun-Wei Tsai, Ting-Wen Liang, Jiun-Huei Ho, Chu-Sing Yang and Ming-Chao Chiang," *A Document Clustering Approach for Search Engines"*, Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, 2006. pp. 126-129
3. Chun-Wei Tsai, Ko-Wei Huang, Ming-Chao Chiang, and Chu-Sing Yang*," A Fast Tree-Based Search Algorithm for Cluster Search Engine"*, Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, October 2009. pp. 8.
4. Daniel Ramage, Paul Heymann, Christopher D. Manning, Hector Garcia-Molina," *Clustering the Tagged Web"*, Proceedings of the SIAM International conference on Data mining, 2008.
5. Hsu*,"Using Topic Keyword Clusters for Automatic Document Clustering",* Proceedings of the 3rd International Conference on Information Technology and Applications, pp. 78-82, 2005.
6. Jun Tang, Xiaojuan Zhao," *An Improved Web Information Summarization Based on SSSC"*, Proceedings of the 2nd International Asia Conference on Informatics in Control, Automation and Robotics, pp. 12-15, 2010.
7. O. Zamir, O. Etzioni, O. Madanim, and R.M. Karp, "*Fast and Intuitive Clustering of Web Documents*," Proceedings of 3rd International Conference on Knowledge Discovery and Data Mining, pp. 287-290, pp. 34, Aug. 1997.
8. M. Steinbach, G. Karypis, and V. Kumar, "*A Comparison of Document Clustering Techniques*," Proceedings of Workshop on Text Mining, pp. 114-116, Aug. 2000.
9. F. Beil, M. Ester, and X. Xu, "*Frequent Term-Based Text Clustering*," Proceedings of 8th International Conference on Knowledge Discovery and Data Mining (KDD 2002), pp. 436-442, 2002.
10. A.K. Jain and R.C. Dubes, "*Algorithms for Clustering Data",* Englewood Cliffs, N.J.: Prentice Hall, 1988.
11. Pavel Berkhin,"*Survey of Clustering Data Mining Techniques"*, Proceedings of the 14th International Conference on Data Engineering Orlando, FL, USA
12. S. Soderland, "*Learning Information Extraction Rules for Semi- Structured and Free Text*," Machine Learning, vol. 34, nos. 1-3, pp.233-272, 1999.
13. S.Z. Selim and M.A. Ismail, *"K-means-type Algorithms: A Generalized Convergence Theorem and Characterization of Local Optimality",* IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 6, pp. 81- 87, 1984.

14. T. Kanungo, D.M. Mount, N.S. Netanyahu, C. Piatko, R. Silverman, and A.Y. Wu, "*The Analysis of a Simple k-means Clustering Algorithm*"º Proceedings of Sixteenth Annual ACM Symposium on Computational Geometry, pp. 100-109, June 2000.
15. K. Alsabti, S. Ranka, and V. Singh, "*An Efficient k-means Clustering Algorithm*", Proceedings of First Workshop High Performance Data Mining, Mar. 1998.