SEARCH ENGINE WITH PARALLEL PROCESSING AND INCREMENTAL K-MEANS FOR FAST SEARCH AND RETRIEVAL

Krishna Kiran Kattamuri[1] and Rupa Chiramdasu[2]
Department of Computer Science Engineering, VVIT, Guntur, India

ABSTRACT

Search Engines are inefficient when there is no optimization. Here, optimization is the process of improving the performance with the focus on time and accuracy. Many techniques are proposed for optimizing the search engines. In this paper, we considered the best techniques existed with certain limitations and now overcoming them with suitable solutions using parallel processing and k -means clustering. In particular, we presented the importance of parallel processing the vector space algorithm for document indexing and incremental updating of means in k-means clustering algorithm in the real world scenario. At the end of this paper, you will have complete implementation details of search engine providing the efficient measures at real time.

KEYWORDS: *Search Engine, Optimization, parallel processing, k-means clustering.*

I. INTRODUCTION

Search engines are the most frequently used to finding specific information on the vast expanse of the World Wide Web. Directory based search engines are very helpful in retrieving information from large human powered databases. It is a process which holds the major stages such as document indexing, stemming and clustering. Document indexing is performed in almost every area of text mining and information retrieval [1]. Vector space model is a widely used technique for efficient indexing of documents. It is used for information retrieval by indexing the documents and query and finding similarity between them. The process is involved in three stages [2]. The first stage is the document indexing where document key terms are extracted from the document text.

The second stage is the weighting of the indexed terms to enhance retrieval of document relevant to the user [11]. The last stage ranks the document with respect to the query according to a similarity measure. In the document indexing, all the keywords are extracted by eliminating common words like is, that, was etc., Thus, all the documents are contained with keywords only. Common words are eliminated by maintaining a stop word list and removing all the stop words in the documents by comparing the document words with that of stop words. After elimination of common words, the remaining terms known as keywords are indexed. All the document keywords are considered in the index by calling it as main index [12]. With respect to every term in main index, the documents are indexed called as a document index.

In the weighting stage, the term weighting for the vector space model has entirely been based on single term statistics [1, 2]. There are three main factors term weighting, term frequency factor, collection frequency factor and length normalization factor. These three factors are multiplied together to make the resulting term weight. A common weighting scheme we used in this system for terms within a document is the frequency of occurrence**.** The term frequency is somewhat content descriptive for the documents and is generally used as the basis of a weighted document vector**.** We generally use the term frequency by inverse document frequency as the weight for the terms in all the

documents. This term weighting is effective in finding the relevant documents with respect to the query.
The similarity in vector space models is determined by using associative coefficients based on the dot product of the document vector and query vector, where word overlap indicates similarity. The dot product is usually normalized. The most popular similarity measure is the cosine coefficient, which measures the angle between the document vector and the query vector [10]. Relevance rankings are calculated by comparing the deviation of angles between each document vector and the original query vector where the query is represented as same kind of vector as the documents. We used the cosine angle similarity measure which calculates the distances between the documents and query. Cosine angle is defined as the dot product of the query vector to the document vector divided by the modulus of the query vector and document vector.

$$\text{Cosine}(q, d) = \frac{(q \bullet d)}{||q||\,||d||}$$

where 'q' represents query vector and 'd' represents the document vector.
The angle measured is the similarity angle for the query to the document. We considered cosine angle because as the angle between query and document increases, the value of cosine decreases [6].

1.1 Stemming and Clustering

Stemming is a process for reducing inflected words to their stem, base or root form. Through stemming, we can able to reduce the dimensionality of occurrence matrix. The major step in optimizing a search engine is done by making entire database in to clusters. We used k-means clustering algorithm for partitioning the entire database in to clusters in the proposed approach. So, in each cluster, there will have similar documents. The documents in one cluster are different from other clusters. Thus, clustering reduces the burden of searching the entire database [8, 12].

II. RELATED WORK

In Vector Space Model, we have the dimensionality problem as there are more words in the documents to index even though common words are eliminated [1]. The system takes more time to index the documents and the complexity increases gradually as the number of documents increases. We used K-means algorithm for clustering the documents [3, 4]. This algorithm divides all the documents in to 'k' clusters in which each document belongs to the cluster with the nearest centroid. But, we need to know the k value initially and probably we stuck at this optimal value. So, we need to find out optimal value of 'k' clusters. For this, we have proposed a new algorithm for addressing this problem.
We have used stemming concept for reducing the dimensionality problem [8]. Stemming is a process for reducing inflected words to their stem, base or root form. In most cases, morphological variants of words have similar semantic interpretations and can be considered as equivalent for the purpose of IR applications. For this reason, a number of so-called stemming Algorithms, or stemmers, have been developed, which attempt to reduce a word to its stem or root form. Thus, the key terms of a query or document are represented by stems rather than by the original words [5, 8]. This not only means that different variants of a term can be conflated to a single representative form – it also reduces the dictionary size, that is, the number of distinct terms needed for representing a set of documents. A smaller dictionary size results in a saving of storage space and processing time. We used porter stemmer as an efficient and widely used stemming algorithm. Porter Stemmer is a context sensitive suffix removal algorithm [9].

III. PROPOSED APPROACH

Vector space model has a serious problem of time complexity. When implemented with vector space model only each and every term in all documents need to be parsed one after the other [5]. This will make the execution of search engine slow. In order to avoid this problem, we need to carefully process the parallel execution of parsing documents for indexing the documents as shown in Fig 1.

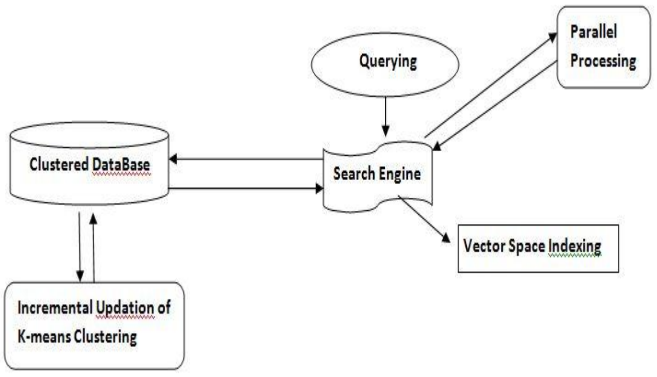


Fig 1. Proposed Frame work

3.1 Parallel Processing Vector Space Model

This is illustrated in a procedure.

1. Consider each document as a thread (separate process).
2. Form main index vector by processing all documents at once. Here, we should verify for duplicate term indexes.
3. Now, form document indexes for all the documents at once.
4. When user given a query, form a query index.
5. Calculate the cosine angle for all documents simultaneously by dot product of query vector and document vector divided by modules of query vector and document vector.
6. Ranking is given based on the angle measure. The more the angle measure value, the more the probability.

The entire process is done in parallel so as to speed up the execution and improve the performance of search engine as shown in fig 2.

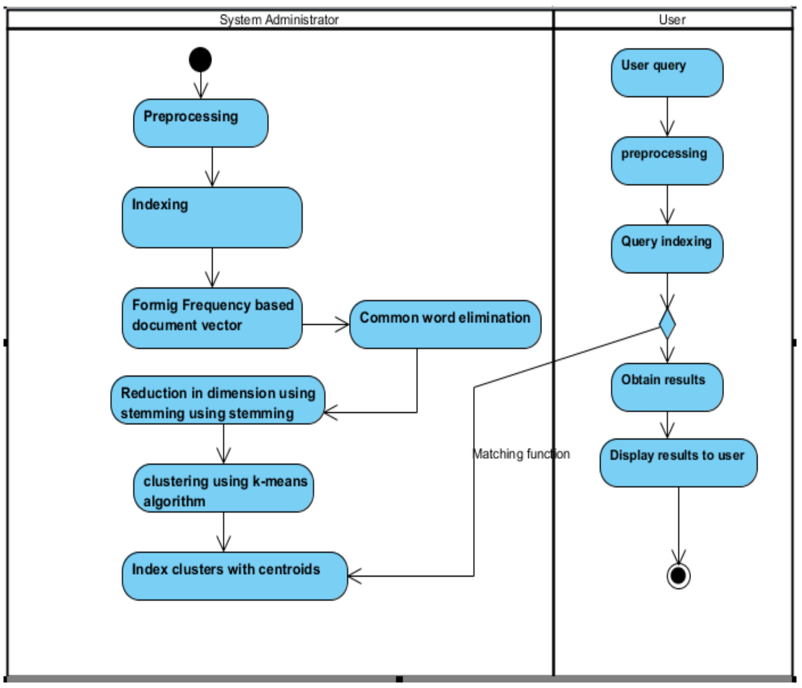


Fig 2. Search Engine process

3. 2 Incremental updating of K-means

Given a set, S, of documents and their corresponding vector representations, we define the centroid vector c to be

$$\text{Centroid, } C = \frac{1}{|s|}\sum_{d=S} d$$

which is nothing more than the vector obtained by averaging the weights of the various terms present in the documents of S. Analogously to documents, the similarity between two centroid vectors and between a document and a centroid vector are computed using the cosine measure, i.e.,

$$\text{Cosine}(\mathbf{d}, \mathbf{c}) = \frac{(d \cdot c)}{||\mathbf{d}||.\ ||\mathbf{c}||} = \frac{(\mathbf{d} \cdot \mathbf{c})}{||\mathbf{c}||}$$

$$\text{cosine}(\mathbf{c1}, \mathbf{c2}) = \frac{(\mathbf{c1} \cdot \mathbf{c2})}{||\mathbf{c1}||\ ||\mathbf{c2}||}$$

Note that even though the document vectors are of length one, the centroid vectors will not necessarily be of unit length.
For K-means clustering, the cosine measure is used to compute which document centroid is closest to a given document. While a median is sometimes used as the centroid for K-means clustering, we follow the common practice of using the mean [9]. The mean is easier to calculate than the median and has a number of nice mathematical properties. For example, calculating the dot product between a document and a cluster centroid is equivalent to calculating the average similarity between that document and all the documents that comprise the cluster the centroid represents. This observation is actually the basis of "intra-cluster similarity". Mathematically,

$$d1\,.c = \frac{1}{|s|}\sum_{d \in S} d1\,.d = \frac{1}{|s|}\sum_{d \in S} \text{cosine}(d1\,.d)$$

Also the square of the length of the centroid vector is just the average pair wise similarity between all points in the cluster [6]. After calculating the first centroid, we calculate the distances to every document with respect to centroid. The greatest distance and least distance between the document vector and centroid vector are identified. Second centroid is obtained by calculating the average between the term frequency of greatest distanced document vector and the least distanced document vector.
Third centroid is calculated by taking the average between term frequency of first centroid and the greatest distanced document vector obtained. Thus, we stop the process of finding centroids and the k value is limited to three. All the documents that are similar to the above centroids are grouped to form clusters. Here, we maintain a threshold value for the number of documents in the cluster. Whenever new documents are added in to the database, we place the documents in to particular cluster by calculating the distance between the document terms and the centroids. We have a threshold value of 15 documents for each cluster. Whenever it exceeds the threshold value, the documents are clustered. All the above steps are again repeated until proper clusters are formed with centroids maintaining a threshold frequency. By this optimized approach of k-means algorithm, we can able to increase the efficiency of clustering and in effect, search engine produces optimal results.

IV. CASE STUDY

We have proposed a directory based Search Engine for journal articles. In this system, we have used various techniques like Vector Space Indexing, Stemming [8], Cosine Measure Ranking and K-means clustering for efficient search and retrieval [1, 7]. With Vector Space Indexing, all the documents in the database will be indexed against terms in each document. The problem arises here as it takes huge time in order to process all the terms step by step. In order to reduce terms for indexing, we used Stemmer technique that optimizes the terms. We have done an optimized way of searching documents by Clustering which reduces the searching time by limiting the irrelevant documents. For this, we used K-means clustering algorithm which is a best algorithm for differentiating clusters.
In this method, we use an initial value of 'k' as a means for clustering. But, finding the value of 'k' accurately will not be possible and that will affect the entire clustering [10]. So, a new technique is proposed where 'k' value is updated in an incremental procedure depending on the number of documents in a cluster. With these techniques, Search Engine will be capable in producing accurate results. For a search engine, a step-by-step procedure of indexing the documents, performing search and producing the results will take effort and time. So, we need a technique which can perform all the steps at a time. The concept of Parallel Processing perfectly works here as it processes all the works simultaneously [13]. It is shown as fig 3.

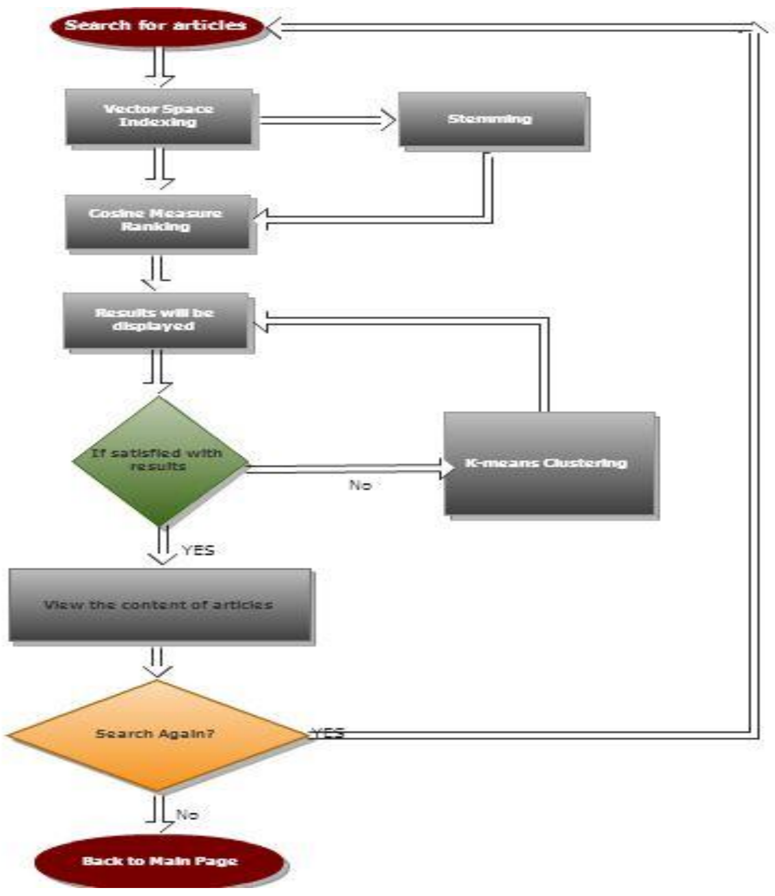


Fig 3. Search Engine with parallel Processing and K-means

V. Results

The implemented Search Engine produces results on parallel processing the search documents and finding correct location of cluster in the database. Fig 4 shows that the user is querying keywords for required journal article in the search box. i. e Mobile marketing.

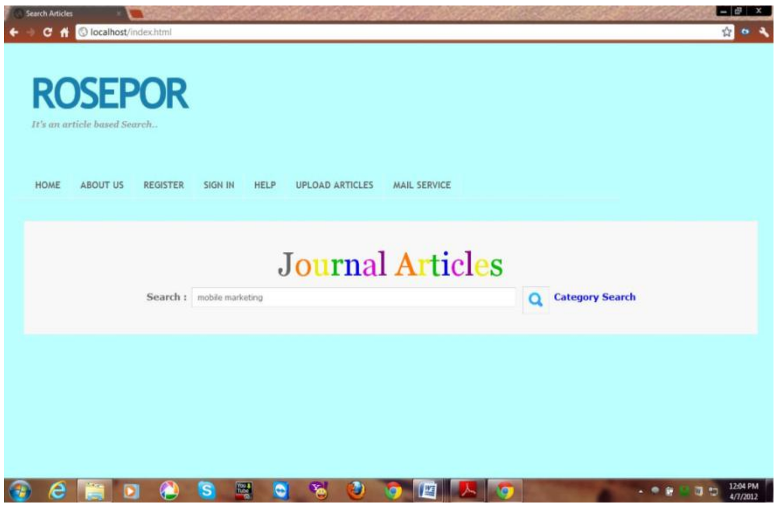
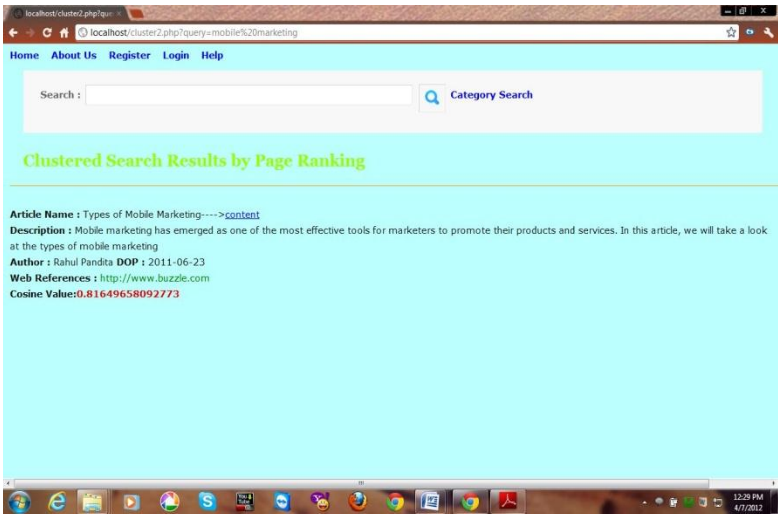


Fig 4. User required journal article name

Fig 5 shows the search results according the keywords in the query box of Fig 4. Here, the search engine finds the exact cluster with the keywords and parallel process the similarity documents with ranking.

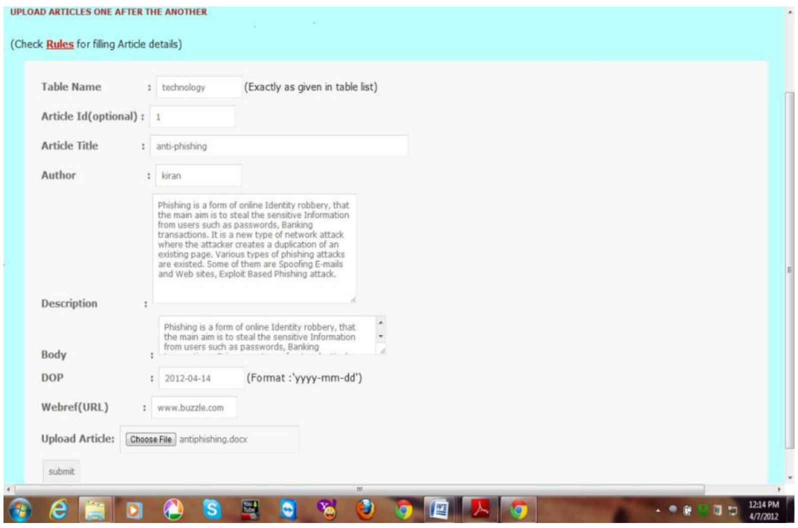
Fig 6 is an uploading application in our proposed search engine. To test the efficiency of proposed techniques we requires abundant database from various resources. For this, we provided an application to upload different journal articles into the database.


localhost/cluster2.php?que
localhost/cluster2.php?query=mobile%20marketing
Home
About Us
Register
Login
Help
Search :
Category Search

Clustered Search Results by Page Ranking

Article Name : Types of Mobile Marketing---->content
Description : Mobile marketing has emerged as one of the most effective tools for marketers to promote their products and services. In this article, we will take a look at the types of mobile marketing
Author : Rahul Pandita **DOP :** 2011-06-23
Web References : http://www.buzzle.com
Cosine Value:0.81649658092773

Fig 5. Search Results for Mobile Marketing



UPLOAD ARTICLES ONE AFTER THE ANOTHER

(Check Rules for filling Article details)

Table Name	: technology	(Exactly as given in table list)
Article Id(optional)	: 1	
Article Title	: anti-phishing	
Author	: kiran	
Description	: Phishing is a form of online Identity robbery, that the main aim is to steal the sensitive Information from users such as passwords, Banking transactions. It is a new type of network attack where the attacker creates a duplication of an existing page. Various types of phishing attacks are existed. Some of them are Spoofing E-mails and Web sites, Exploit Based Phishing attack.	
Body	: Phishing is a form of online Identity robbery, that the main aim is to steal the sensitive Information from users such as passwords, Banking	
DOP	: 2012-04-14	(Format :'yyyy-mm-dd')
Webref(URL)	: www.buzzle.com	
Upload Article:	Choose File antiphishing.docx	

Fig 6. Uploading the new paper

VI. Conclusion and Future Scope

In this paper, we have focused on two different techniques combined for fast search retrieval. It has provided a framework for parallel processing the search engine and procedure for incremental updating of K-means. We adopted these techniques because of limitations in finding initial k-means and retrieving time. Parallel processing now speeds up the execution of search engine and k-means clustering in reducing the search time. The implementation of a search engine with k-means clustering and parallel processing is tested with several queries with different data sets. The results obtained in the tests proved that the above approach is efficient.
As Search Engines are implemented on large scales of databases, the efficiency of retrieving must be verified with such databases. In future, we test the query results of hyper-textual web databases in real time. We are also interested in multi core architectures implementing query optimization, cache updating and many other multi-threading techniques for efficient searching.

References

[1]. David A. Grossman, OphirFrieder, (2004) "Information Retrieval: algorithms and heuristics" Second Edition

[2]. G. Salton, A. Wong, C. S. Yang, (1975), "A vector space model for automatic indexing, Communications of the ACM", v.18 n.11, p.613-620.
[3]. Michael Steinbach,George Karypis and Vipin Kumar,"A Comparison of Document Clustering Techniques" , IEEE
[4]. HinrichSchutze and Craig Silverstein, "Projections for Efficient Document Clustering", SIGIR '97, Philadelphia, PA, 1997.
[5]. K. Goda, T. Tamura, M. Kitsuregawa and A. Chowdhury, "Query Optimization for Vector Space Problems",IEEE.
[6]. DIK L. LEE ,"Document Ranking and the Vector-Space Model", IEEE Transactions
[7]. D. Boley, M. Gini, R. Gross, E.-H. Han, K. Hastings, G. Karypis, V. Kumar, B. Mobasher, and J. Moore. "Partitioning-based clustering for web document categorization. Decision Support Systems" (1999) 27(3):329–341.
[8]. Frakes, W. B. (1992) "Stemming algorithms, Information retrieval: data structures and algorithms", Prentice-Hall, Inc., Upper Saddle River, NJ
[9]. Marie-Claire, J. and Smith, D. (2005), "Conservative stemming for search and indexing", Springer.
[10]. Kanungo, T.; Mount, D. M.; Netanyahu, N. S.; Piatko, C. D.; Silverman, R.; Wu, A. Y. (2002), "An efficient k-means clustering algorithm: Analysis and implementation". IEEE Trans. Pattern Analysis and Machine Intelligence pp: 881–892
[11]. D.B. Cleveland and A.D. Cleveland (2001): "Introduction to indexing and abstracting". 3rd Ed. Englewood, libraries Unlimited, Inc. ISBN 1-56308-641-7, pp105
[12]. Zobel, J., Moffat, A.: Inverted files for text search engines. ACM Comput. Surv.38(2) (2006) 6
[13]. Jia, X., Trotman, A., O'Keefe, R., Huang, Z.: Application-specific disk I/O optimization for a search engine. In: PDCAT '08: Proceedings of the 2008 Ninth International Conference on Parallel and Distributed Computing, Applications and Technologies, Washington, DC, USA, IEEE Computer Society (2008) 399–404

AUTHORS

Krishna Kiran Kattamuri is a student of B.Tech in the field of Computer Science from VVIT, Andhra Pradesh. He has presented papers in the prestigious conferences which are conducted by IEEE, Andhra University. He is very passionate on R & D work. His main research interest includes Information Retrieval, Computer Networks and Information Security.

Ch. Rupa is working as Associate Professor in VVIT, Andhra Pradesh, INDIA. She has received B.Tech from JNTU, Hyderabad, M.Tech (CSIT) and Ph. D (CSE) degrees are from Andhra University. This author became a Life Member of CSI, ISTE, IAENG, IEI, and IACSIT. She published more than 35 papers in various journals and conferences. JNTU kakinada had awarded her as a Young Engineer of 2010. IEI awarded her as National young Engineer of 2011 Govt of A. P and IEI by combined awarded her as Young Engineer of 2012. Her main research interest includes Information Security, Image Processing, and Security algorithms.