2009

Hybrid Clustering with Application to Web Pages

Ameya Sabnis
San Jose State University

HYBRID CLUSTERING WITH APPLICATION TO WEB PAGES

A Writing Project

Presented to

The Faculty of the Department of Computer Science

San Jose State University

In Partial Fulfillment

Of the Requirements for the Degree

Master of Science

by

Ameya Vikas Sabnis

December 2009

APPROVED FOR THE DEPARTMENT OF COMPUTER SCIENCE

Dr. Teng Moh,Department of Computer Science Date

Dr. Mark Stamp, Department of Computer Science Date

Dr. Robert Chun, Department of Computer Science Date

APPROVED FOR THE UNIVERSITY

San Jose State University

ABSTRACT

HYBRID CLUSTERING WITH APPLICATION TO WEB PAGES

by Ameya V. Sabnis

This project explains the process of clustering web pages. With the immense increase in the number of web pages available on the internet, it has become difficult to search for web pages. The clustering of web pages will improve the presentation of web pages to the user and saves the time spent on searching web pages. Various clustering techniques have been proposed by various research scientists to cluster the web pages, but all the techniques suggested have some drawbacks. Since there is lot of scope for further improvement in the field of clustering, the system proposed in this report takes the clustering of web pages a step ahead. The proposed system use the queries from the user and get the results from search engine, then processes the results and provides the final result clusters to users.

ACKNOWLEDGEMENTS

I would like to express my gratitude to Dr. Teng Moh for being my advisor. His guidance and motivation proved invaluable without which the completion of this project would have not been possible.

I would also like to thank my committee members, Dr. Mark Stamp and Dr. Robert Chun, for their time and inputs provided for this work.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. Introduction

The birth of internet is really a gift to the mankind. In the recent years the growth and popularity of the internet has increased to such an extent that every person knows about it and uses it for various purposes. Some people use internet to know new things, while others use it as a means of entertainment. The use of internet is not only limited to the entertainment but it can also be used to conduct research related to work or study, get latest news etc. With each passing movement millions of web pages are added to this internet. The survey conducted by Google in 2008, [21] brought us to the conclusion that there are one trillion unique url's on the internet. The implementation of search engines on the internet made the process of searching some of the topics very easy. Querying the search engine for any particular topic would retrieve the results from the internet and those results are then presented to the users. But since there are many pages on the internet the results obtained by the search engines are also vast. It becomes really difficult for the user to get the particular page from the search engine. If it happened that the Page Rank of the particular page is high then it can be found on the first page of the search engine results, else it can be found at the end of the results. This results in the loss of time for the users as they had to spend the time looking for the particular required page. To overcome the drawbacks of the search technique, it is necessary that the search results are clustered.

Clustering will help to group the similar pages together and the dissimilar pages are not grouped. Presenting this grouped results to the user will help the users to get all the

related pages to their query and also will reduce the time spent by them in searching the related page. Presently there are various recommendations and techniques to cluster the web pages. This report proposes one of the clustering systems which cluster the web pages by taking in the user query.

1.1 Overview of the Project

Clustering is the process of supervised or unsupervised classification [22] of patterns or similar items in group. The data items within the group are similar to each other while the data items in the different group are dissimilar to each other [22]. The process of clustering can be applied to the field of pattern recognition, document analysis, data mining, image segmentation and mathematical programming. A lot of research is being conducted in the field of clustering, to improve the quality of clusters and to name those clusters appropriately. With the number of pages on the internet is increasing day by day it has become immensely important to cluster the web pages. Though many research scientists have worked in this area, but still there is a scope to improve the result with respect to quality and also with respect to the processing time.

The system proposed here in this report combines the two clustering techniques namely K-means clustering technique and Evolutionary Particle Swarm Optimization (EPSO) clustering technique. K-means clustering is partitional clustering algorithm while EPSO came from the Particle Swarm Optimization. Both these techniques have their own advantages and disadvantages. The system implemented here uses the combination of advantages of both the algorithms. The system is first implemented on the simple dataset

which are just co-ordinates on graph, and then this system is applied for the web pages to perform web pages clustering.

1.2 Organization of the Report

This report gives a detailed description about the research work and the experiments carried out. Section 2 explains the basic concepts of the proposed system which are clustering process, clustering types, need of optimization techniques. The section 3 gives an overview of the work that is done till now in the area of clustering particularly hybrid clustering. Section 4 describes the differences between the existing system and the new proposed system. It also explains how the new proposed system will be better than the existing system. This section is basically related to how the motivation for the project was generated. Section 5 deals with the proposed algorithm. It explains the implementation of new algorithm. It also discusses the experimental analysis and results of the proposed algorithm. Section 6 describes the software architecture of the proposed system in detail for web page clustering. It explains various components of the proposed system. Section 7 contains the experimental setup details and how the experiments are performed. This is followed by the explanation of experimental results. Section 9 gives the conclusions and future work that take the area of clustering of web pages up one more level.

2. What is Clustering and Optimization Techniques?

To make the process of clustering simple and efficient, it is necessary that the process of clustering should always be combined with the optimization techniques. The clustering process finds the local results while the optimization process finds the best results among them. Depending upon the data requirement and the clustering process, the optimization methods can be used before or after the clustering process.

2.1 What is Clustering?

The process of forming the group of similar items is known as clustering. The process of clustering can be used in various fields such data clustering, document clustering, web clustering, etc. The process of clustering is very simple and straight forward. Given a certain data points, consider some of the data point as the centroid and calculate the distances of other points with respect to the chosen centroid. Putting the certain threshold to on the maximum distance, the data points which are within the threshold will gel with the respective centroids and the clusters are formed. The total number of clusters formed, depends upon the initial number of centroids selected for clustering. The simple graphical representation of clustering can be shown here.

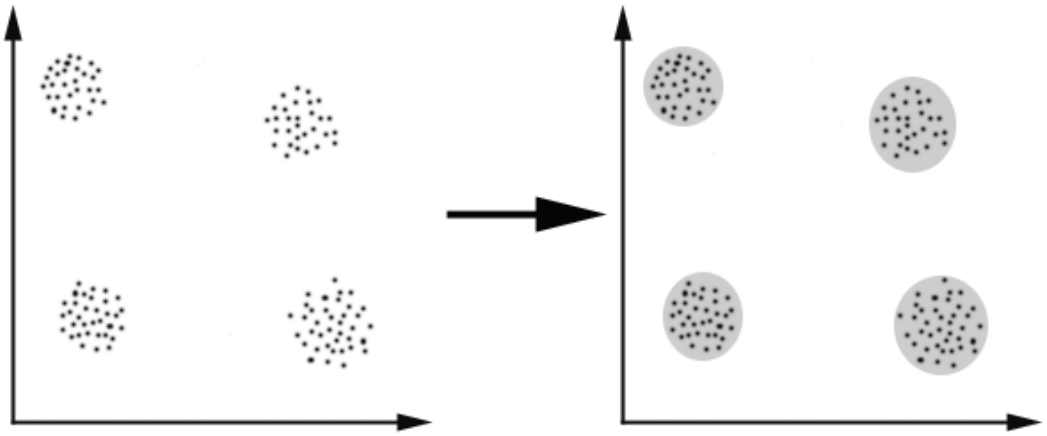

Figure 1: Simple Graphical Representation of Clustering

There are various types of clustering algorithms. Every type of clustering algorithms has got some advantages and some disadvantages. The main types of clustering algorithms are Partitional clustering algorithms, Hierarchical clustering algorithms and Density-based clustering algorithms. The following algorithms are described here:

- **Partitional clustering algorithms**: Partitional clustering algorithms directly try to decompose the dataset into a set of disjoint clusters. The criteria involved to partition the dataset is to minimize some measure of dissimilarity in the samples within the cluster. The algorithm goes on assigning data to the cluster depending upon its closeness to the centroid of the cluster.[24]
- **Hierarchical clustering algorithms**: Hierarchical clustering algorithms create the hierarchy or tree view of clusters also known as dendrogram. The working procedure of this algorithm is simple. First each data point is considered as an individual cluster. Then the merging operation takes place. Depending upon the

similarity of the data point's two individual clusters merged. The process is continued until no data point can be merged.[25]

- **Density-based clustering algorithms**: Density-based clustering algorithm, cluster the data depending upon the normal density distribution technique. This algorithm applies a local cluster criterion. Clusters are regarded as the region in the data space where the concentration of points is high and are separated by the region of low object density [23].

2.2 What are Optimization Techniques?

The Optimization is the method of choosing the best result out of the available results. There are various optimization techniques used in the field of mathematics and computer science. Each optimization technique has its own advantages and solving properties. Many optimization techniques are used along with the clustering algorithms. Following are the main optimization techniques used with the clustering algorithm, Ant Colony Optimization [4], Harmony Search [5], Particle Swarm Optimization [7, 8, 9], Tabu Search [6], etc. A general diagram showing the clustering process along with the optimization methods is given below. The description for each blocks is given after the diagram.

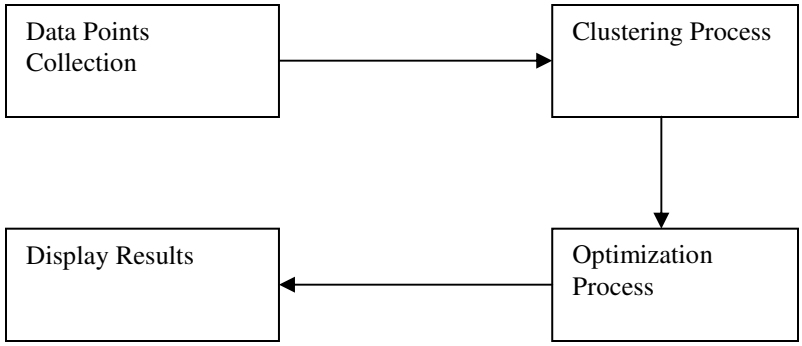

Figure 2: High Level Steps of Clustering and Optimization

- Data Points Collection: It is very much necessary that the data points are collected first before beginning with the clustering. The data points can be anything, like graphical points or the text documents or the web documents. These data points are then passed to the clustering process.
- Clustering Process: The clustering process takes in these data points and performs clustering on them. Depending upon the similarity of the data points clusters are generated. These clusters are then given to the Optimization process.
- Optimization Process: Optimization process takes in the clustering results. Clustering results are generally raw results and never represents the best result. Optimization techniques are used to get the best result out of the optimization process. The result is then given to the display process.
- Display Results: In this step the results are labeled properly and those clusters are then displayed to the user. If these data points are the web pages then the links for the pages clustered are displayed to the users.

3. Previous Work

This section describes the work that is previously performed by the research students and scientists in the field of optimization and clustering. Research to improve the quality of the clustering results is going on for several years. The main focus of these researches is to form the accurate clusters and represent it to the users in the meaningful way. With the inclusion of the optimization techniques along with the clustering algorithms there is much more scope to form most suitable clusters.

The basic of clustering algorithms is explained in [1]. This book explains all the necessary components for the data clustering. It also explains the various algorithms which are used for clustering. The review about the data clustering is explained in [2]. In this survey paper the authors have explained everything about the clustering algorithms. It also explained the hierarchical and partitional clustering algorithms process. This paper also explains the disadvantages of the hierarchical clustering algorithms. Though hierarchical algorithms are considered as the best clustering algorithms [3], there is no way we can reallocate the datasets which are poorly clustered in previous stages. Hence partitional algorithm (K-means algorithm) is considered to be superior over hierarchical algorithms [3]. Along with the clustering algorithms it is necessary that we use some of the optimization techniques to get better results.

In [4] the authors explained the process of combining the Ant Colony Optimization with the K-means algorithm. The hybrid algorithm was used to solve the image clustering problem. The paper [5] explains the novel algorithm for clustering of web pages. The algorithm uses the combination of K-means algorithm and the Harmony Search

algorithm. The algorithm first performs the K-means clustering algorithm and then once it gets the intermediate results, these results are then processed by the Harmony Search algorithm. The authors in this paper [6] explain the process of clustering using the Tabu Search approach. This approach is used for the pattern recognition technique using the process of clustering.

The papers [7, 8, 9] explains the process of Particle Swarm Optimization for clustering. In particle swarm optimization the particles are considered as agents flying through the given data space. When the particles fly through the space, the each updated position of those particles in considered as the solution. In this particular concept the problem could be the search of best available position. Each position of the particle is considered to be the one of the solution. A particle's location in multi-dimensional space can be considered as one of the solution to the problem. When the particle moves to the new location the new solution is generated. These solutions are evaluated by the fitness function. Various equations are also considered in this technique. The local best fitness value and the global best fitness value are calculated regularly for each particle. The papers [3, 10] explains the process of clustering of data using the hybrid algorithm. The hybrid algorithm is nothing but the combination of K-means and Particle Swarm Optimization (PSO) algorithm. In this hybrid algorithm the PSO is considered as the optimization technique to find the best local results. The results obtained by the hybrid algorithms were much better when compared with the results obtained by the Particle Swarm Optimization technique alone. There are many papers related to swarm optimization and clustering. [3, 7, 8, 9, 10]

Evolutionary Particle Swarm Optimization (EPSO) explained in the paper [11] refers to the new technique of swarm optimization. The author uses the concept of evolution to cluster the data particles. Normalized Distribution technique is used to consider all the points available in the vector space. In Evolutionary Particle Swarm Optimization the swarm is taken as the cluster solution. Then particles are merged to optimize the swarm size into optimal number of particles each representing an individual cluster centroid with associated data vectors. After doing more research, it was found that there is a scope to improve this algorithm. The EPSO algorithm is linear in time complexity, which means that with increase in number of particles, the execution time increases. One of the ways to improve this algorithm is hybridized it using the K-means algorithm.

After setting down the clustering algorithm it is necessary to know the process of the clustering of the web pages. It is required to know the format of the web pages in which they are presented. These web pages can be in HTML format or XML format. Generally web pages are the plain text documents with the HTML / XML tags embedded into it, to make the data on the page look attractive. The paper [19] explains the process of web clustering engines. The paper explains the basic flow of the clustering engines. The main work in this clustering process is data acquisition. Once the web data is acquired, then this data is processed and cleaned. Cleaning refers to removing the tags, images, multimedia documents. After this the only item kept from the web pages data are the plain text documents. These plain text documents are appended with the document ID, so that they can be recognized easily. Once this is done the plain text documents are tokenized. The tokenization takes place document by document.

The labeling of clusters in explained in paper [18]. It explains the general approach of labeling the clusters. The paper [20] also explains the process of clustering of web pages with different concept and approach. Here the authors form the label first and then performs the clustering. The labeling is formed by determining the number of occurrences of the terms in the document. Once the labeling is done results are presented to the user.

4. Motivation for Proposed System:

After thoroughly researching on the Evolutionary Particle Swarm Optimization (EPSO) algorithm, it was found that the EPSO algorithm can work along with the K-means algorithm. The EPSO algorithm considers each data particle as an individual swarm. Depending upon the strength of the swarm it decides whether to form the cluster of the swarm. The EPSO algorithm is based on the idea of the generation based evolution of the swarm. The swarm evolves through different intermediate generations to reach a final generation. Particles are initialized in the first generation and after each generation the swarm evolves to a stronger swarm by consuming the weaker particles of that generation by the stronger ones. The stronger the particle is, the greater its chance of survival to the next generation. Stronger particles make mature and stable generations.

Consider the example where there at generation (x) there are four swarms S1, S2, S3, S4. Of these four swarms let us consider that the strength of swarms S2 and S4 is weak. Then in this case it is more likely that the swarms S1 and S3 will consume the weak swarms to get more strength and gets evolved into next generation. This process

continues unless a stable strength is reached. Following figure shows the above scenario of evolutionary particle swarm optimization.

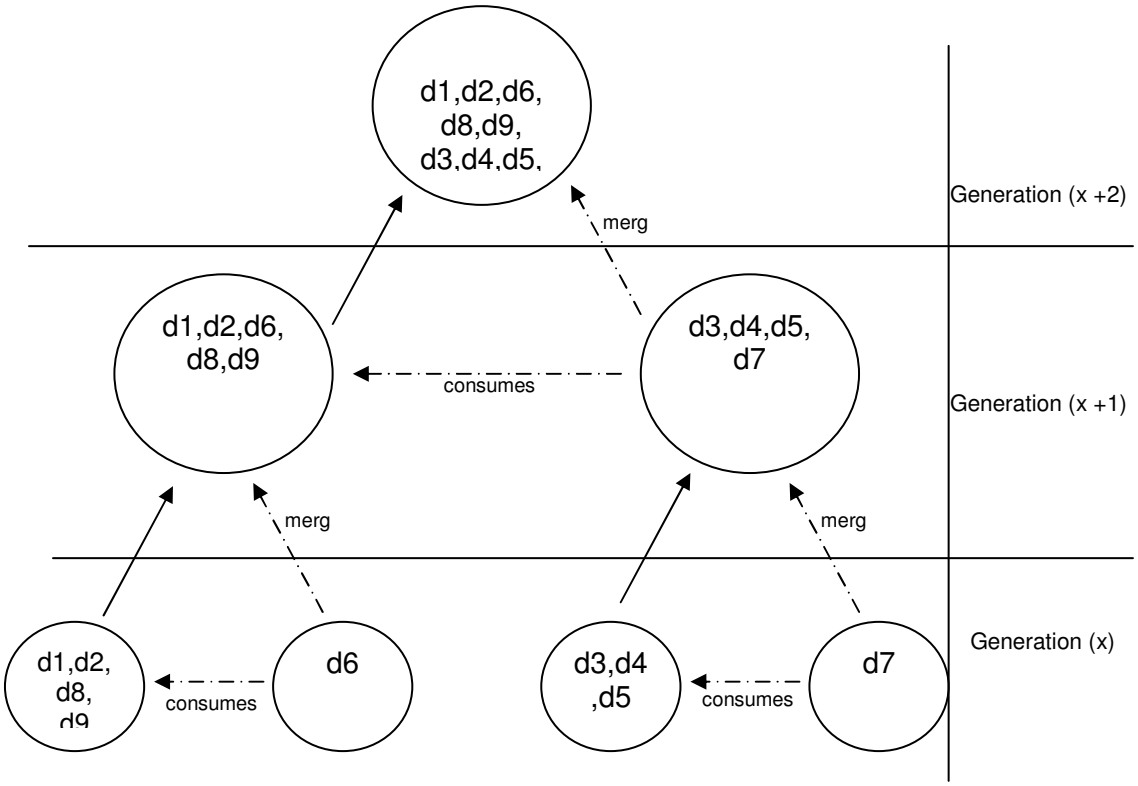

Figure 3: Concept of Evolutionary Particle Swarm Optimization

The above figure 3 illustrates the concept of Evolutionary Particle Swarm Optimization. It shows with the evolution of next generation the weak swarms are consumed by the strong swarms. Once the weak swarms are consumed the generation of the swarm evolves to the next generation. The swarming process of EPSO-clustering starts the first generation of the swarm by initializing the particles to the data vectors from the input data in a uniform manner so that they cover the whole input space. During the first iteration the particle updates the position regularly. After the predefined iteration the next generation starts. This generation will have better particles after the weaker

particles are consumed by the nearest stronger particles in the previous generation. The pseudo code for the EPSO algorithm is given below.

Step 1: Initialization of Particles

a. Initialize $V_i(t)$, $X_i(t)$, V_{max}, q_1 and q_2
b. Initialize swarm size and generation
c. Initialize particles to input data

Step 2: Iterate Generation

a. Iterate Swarm
    i. Find winning data vectors
    ii. Update Velocity and Position
b. Evaluate the strength of Swarm
    i. Iterate Generation
    ii. Consume weak particles
    iii. Recalculate the position

Step 3: Exit on the number of generations fulfilled or stopping criteria reached

In the above given algorithm, $V_i(t)$, V_{max} are the initial velocity and the maximum velocity of the particles. Both these velocities have been defined at the start of the algorithm. $X_i(t)$ is the position of the particle at the time of initialization. The position of each particle is defined at the start of the algorithm. , q_1 and q_2 are the constants necessary for the EPSO algorithm. Along with this the maximum swarm size is also defined. The number of generations is defined as the stopping criteria. In this algorithm the stopping criteria in this algorithm is considered to be the number of clusters formed at the end of

the EPSO process. The particles are also initialized according to the input data. After the iteration of generation and the iteration of the swarms, the winning data particles are calculated. Depending upon the strength of particles the win-lose situation of the data particles is considered.

Though the clustering results obtained by the EPSO algorithm are satisfying the time consumed by the process is very long, since the algorithm works with the linear time complexity. The main drawbacks of the EPSO clustering algorithm are its time complexity and non detection of outliers. One of the ways to make this algorithm work faster is to use the K-means and get the clusters. This process of hybrid clustering will surely help to minimize the time spent on clustering using the EPSO alone.

5. Proposed K-EPSO algorithm:

After the initial research with the Evolutionary Particle Swarm Optimization (EPSO) which takes longer time to process the data sets, the processing time of this algorithm can be reduced further if this algorithm is joined with the K-means algorithm. The K-means algorithm is first implemented, and then the EPSO algorithm is implemented. The main use of K-means algorithm is to get the clusters very quickly and once this is done pass this clusters to the EPSO algorithm. Also the drawback of the outlier points is removed out since K-means automatically discards the outlier points. The pseudo code for the K-EPSO algorithm can be given as

Step 1: Select K-points as initial centroids (K = 10 to 15)

Step 2: **Repeat**

a. Form K-clusters by assigning each point to its closest centroid

b. Recompute the centroid of each cluster

Step 3: **Until** centroid do not change

Step 4: Initialization of Clusters (Particles)

a. Initialize $V_i(t)$, V_{max}, q_1 and q_2

b. Initialize swarm size and generation

c. Initialize clusters to input data

d. Obtain the original position

Step 5: Iterate Generation

c. Iterate Swarm

i. Find winning data vectors

ii. Update Velocity and Position

d. Evaluate the strength of Swarm

i. Iterate Generation

ii. Consume weak particles

iii. Recalculate the position

Step 6: Exit on the number of generations fulfilled or stopping criteria reached

In this algorithm the value of K in the K-means algorithm is user defined. The user has to input the K initial points in the algorithm. It is necessary for the users to know the dataset before using the K-means algorithm. After K-means algorithm, the clusters which are formed are given to the EPSO module. The velocity and the maximum velocity of the

particles are defined. Also the size of swarm and the number of generations is also defined. Then the winning points are determined.

5.1 Implementation of Proposed K-EPSO algorithm

The K-EPSO algorithm which is proposed earlier is implemented using JAVA with Eclipse IDE. The algorithm starts with the K-means clustering algorithm. Initially the value of K in the K-means algorithm is taken very high. In the experiments conducted the value of K in the K-means is considered between 10 and 15. The main reason to consider such a high value for the K-means cluster is to get more number of particles. If we select K to be equal to four or five then the clustering process is done alone by the K-means algorithm and the EPSO algorithm is not utilized.

As the K number of initial points (centroids) are selected the K-means algorithm starts processing this is the first step of the K-means clustering algorithm. In the second step, the remaining points are assigned to the initial centroids. After the points are assigned to the centroid, the centroid is then updated [12]. In the next step the remaining points are assigned to the updated centroid and the centroid is recomputed again by considering all the old and new points. This process goes on until no more change occurs with the updated centroid.

At the end of K-means algorithm the working process of the goes to the second phase that is Evolutionary Particle Swarm Optimization (EPSO) algorithm. In the first step of EPSO phase, initialize the Velocity of particle ($V_i(t)$), initialize its maximum velocity (V_{max}) and also initialize the other parameters necessary for the EPSO phase. Initialize the swarm size and swarm generation, obtain the original position of the particle and also

initialize the particles. In the second step iterate the generation of swarms. Obtain the winning data points and update the velocity and the position. Along with this evaluate the strength of the swarm, determine which swarms are weak and which swarms are strong. In this phase the strength of the swarms is determined by the number of data points it contains. If one of the swarm consists of only two data points while the other has more than two points, then the second swarm consumes the first swarm. After the weak particles are consumed then the position and the velocity of the particles are again calculated and updated accordingly. The algorithm terminates when it reaches the stopping criteria. The stopping criteria can be the number of generations reached or the number of clusters formed.

Once the stopping criteria is reached the clusters are obtained, the average distance for all the points within the clusters is calculated. To verify that the K-EPSO hybrid algorithm performs similar to the EPSO algorithm, the experiments were conducted on the standard clustering data. The experimental setup and its analysis is explained in the next part of the report.

5.2 Experimental Analysis of K-EPSO algorithm

The experiment was conducted using the data similar to the benchmark standard clustering dataset [11]. The dataset has two attributes with 75 data points and has four natural clusters. This dataset has been used in many clustering algorithms to check the quality of the clusters formed.

The parameters for the K-means algorithm are kept constant throughout the experiment. For better results it is assumed that the value of K, in the K-means algorithm

should be as big as possible. By keeping the value of K to be large, enough clusters can be given to the EPSO module. Also some of the outliers are discarded from the K-means algorithm, thus making the data set as pure as possible. The parameters for the Evolutionary Particle Swarm Optimization algorithm are also kept almost similar as proposed in paper [11]. Some of the parameters such as the maximum velocity, number of iterations, generation threshold are most influential parameters when performing the EPSO experiments. In the proposed algorithm all the parameters are fine tuned to make this algorithm run in optimized condition.

The performance of the algorithm is tested against the K-means algorithm and EPSO algorithm. The centroid selected in the K-means algorithm was kept same in the K-Evolutionary Particle Swarm Optimization algorithm. The results proved to be satisfactory and the time taken by the proposed algorithm is less when compared with the time taken by the Evolutionary Particle Swarm Optimization algorithm. The performances of all the algorithms are depicted in Table 1.

The strength of clusters found in each algorithm is tested using the average intra cluster distance. This is the standard metric used to determine the quality of the cluster. The more, small the ADDC value is the more compact is the cluster formed during the clustering process. Also average distance from centroid is calculated in all the algorithms.

TABLE I K-means and EPSO and K-EPSO clustering comparison results

Method	No. of Clusters	Cluster	Number of Data Vectors	Avg. Distance form Centroid	Execution time
K-means	4	1	15	8.996	2.54 secs
		2	17	13.920	
		3	20	12.432	
		4	23	10.549	
EPSO Clustering	4	1	14	9.008	60.21 secs
		2	17	13.639	
		3	20	12.752	
		4	24	10.581	
K-EPSO Clustering	4	1	14	9.008	38.18 secs
		2	17	13.639	
		3	20	12.752	
		4	24	10.581	

The table 1 also shows the execution time required by all the algorithms. Every algorithm produces four clusters with different number of data vectors in each cluster. The average distance from the centroid, gives us the idea that EPSO and K-EPSO forms much compact clusters than the K-means algorithm. K-means algorithm known for its speed performs the clustering very quickly; hence the execution time for the K-means algorithm is very small. Time taken by the EPSO algorithm is much more than the time taken by the K-EPSO algorithm. If we calculate the gain in time, it can be found that there is nearly 30% saving in the execution of the K-EPSO algorithm. The results obtained by both EPSO and K-EPSO clustering algorithms are same. Figure 4 shows the convergence graph for all the algorithms. Though K-means executes faster, the quality is not maintained and every time it is necessary for the users to know the dataset. Also the

input for the initial centroids is requirement in K-means. With change in the initial centroid the quality of cluster changes.

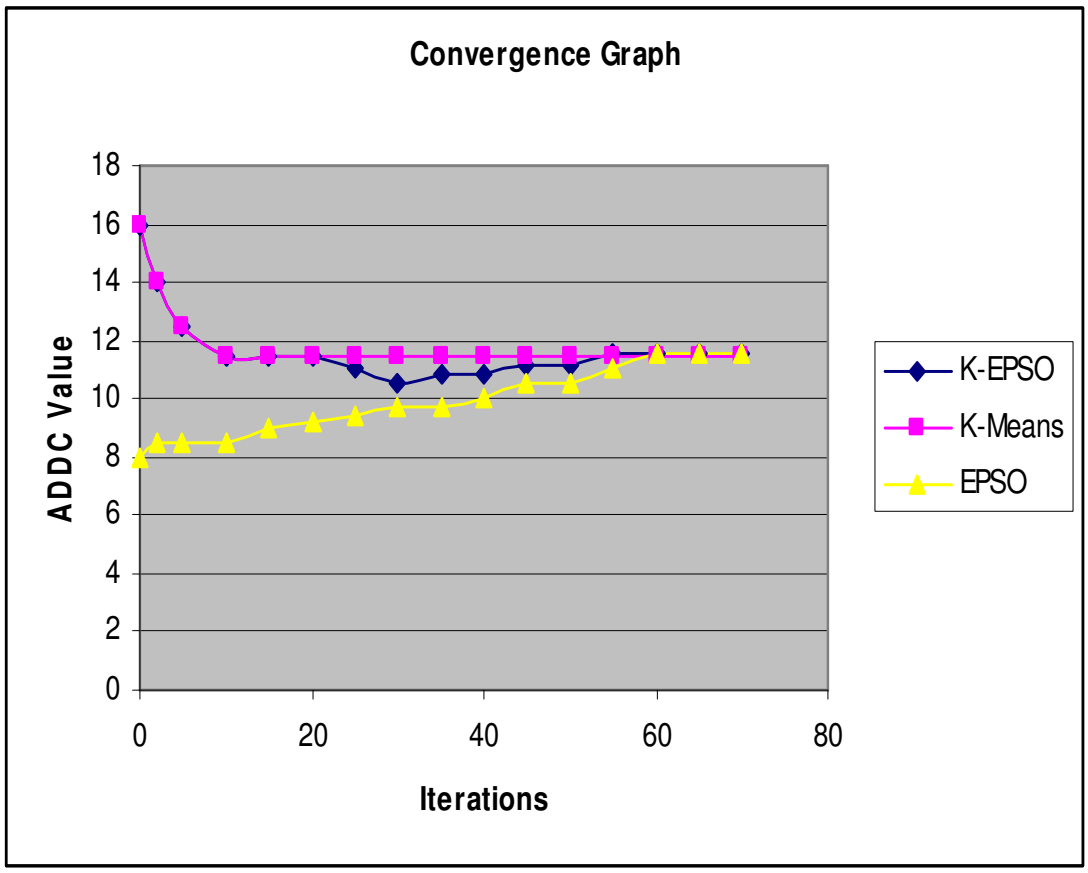

Figure 4: Convergence Graph of all Algorithms

From the figure it can be seen that the Average distance for all the algorithms is similar. K-means algorithm works for first fifteen iterations and then it gives the constant Average distance from the centroid of the cluster (ADDC) value. EPSO starts with the lower ADDC value, since in EPSO all the particles are considered to be individual clusters, the ADDC value at the start of the EPSO algorithm is less as compared to the other algorithms. The K-EPSO starts with the K-means algorithm first and hence the

ADDC value for the K-EPSO algorithm is similar to the values of K-means algorithm and EPSO algorithm.

6. Software Architecture of the Proposed System:

The software architecture of the proposed system is shown below:

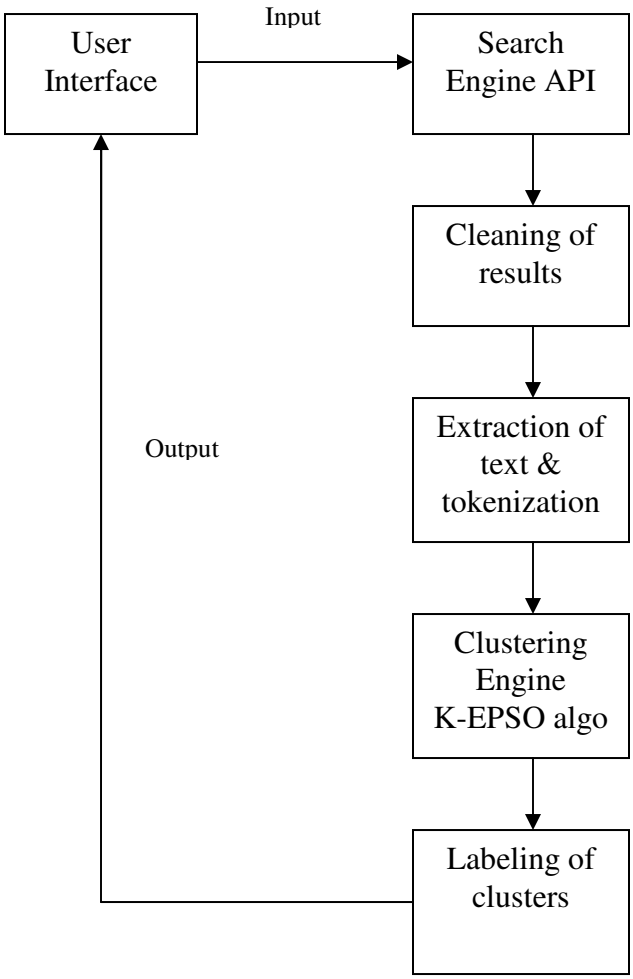

Figure 5: Proposed Software Architecture

The figure above gives the software architecture for the proposed system. The user enters the query, which is read by the search engine API. The search engine API retrieves

the search results, these retrieved results are cleaned and tokenized and then given to the K-EPSO clustering engine. The last stage is to generate labels to the clusters. After the label generation, the result is displayed to the user. Here is the complete description of the architecture.

User Interface:

The user interface is nothing but the aggregate means by which users interacts with the system [14]. The user interface provides the means of input, where user can submit the query and it also provides the means of output, where users can see the results of the query submitted. In the architecture above the user query would be any search term, which the user needs to find on the internet.

Search Engine API:

API's are the kind of interface implemented by the software program in order to allow other software to interact with it [13]. API's are implemented by applications, libraries or operating systems to define how other software can make calls to or request services from them. The search engine API provides the interface to take in the search query. Once the user enters the query through the user interface, the software running the user interface requests the service of the search engine. Various protocols and conventions are considered to make the request for the service successful. The search engine API's takes in the query from the user interface and then passes it to the search engine. The search engine performs the normal searching operation and it returns the results.

Once the results are retrieved, they are again passed through the API and then displayed to the users. The results are generally in XML or HTML format. These files need to be processed to remove the tags. Removal of tags will lead us to the plain text results which can be used for the process of clustering.

Cleaning of Results:

Generally the web page consists of the text data and the multimedia data. For the purpose of clustering only the text data is considered and the multimedia data is just ignored. After we get the results from the search API, it is necessary that the results should be cleaned. This means that it is necessary that the images and all other related stuff from the web page must be discarded and only the text should be considered. The unnecessary HTML and XML tags should be removed. This process is done using the XML parser. Generally the output of the search engine API is the XML file and hence to process this file and to obtain the text material XML parser is used. The parser removes all the tags and just keeps the plain text for the purpose of processing. The XML parser helps to separate the titles, urls, texts on the web pages.

Extraction of text and its tokenization:

The text documents are not easy to cluster. The process is much more complicated when compared to the standard data point clustering process. Once we get the documents these are nothing but the data from the web pages we need to determine the similarity of the document for the text clustering. The book [15] explains all the techniques and procedures and processes required for the text clustering. For the text clustering it is essential that the frequency of the terms appearing is the document is calculated. The

higher the frequency of the term appearing in the document, maximum is the percentage of which the document is related to that term. Consider the following example where person gets the following terms computer, repairing, motherboard, hard drives, etc if the frequency of these mentioned terms is high then one can easily conclude that the web document is related to the computer repair shop or computer sales shop. While calculating the frequency of the terms appearing the document, care it taken that the prepositions, conjunctions, adverbs, verbs are avoided. These terms form the stop words. It is most likely that considering the stop words in the process of clustering will definitely lead us to wrong results. The reason to discard these stop words is, the frequency of these stop words in a document is very high. If these are not discarded they will play role in calculating the inverse document frequency which will directly affect the cosine similarity index and thereby the clustering results.

To determine the similarity of the document it is necessary to calculate the term frequency-inverse document frequency (tf-idf). The tf-idf is the weight which is often used in the process of text clustering [16]. This weight is also used to determine the importance of the particular term in the document. This process is carried out in the following way. First the document is processed to remove any of the stop words appearing in the document. Then the terms appearing in the document are counted. If the term appears more than once in the same document, the counter to count its appearance is incremented every time the term is processed. Once this is done a term-document matrix is formed which keeps the track of all the documents and all the terms. The following figure shows the model of the term-document matrix.

	D1	D2	D3	D4	D5	D6	D7
applications	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
binary	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000
computer	1.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000
engineering	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000
eps	0.0000	0.0000	1.0000	1.0000	0.0000	0.0000	0.0000
generation	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000
graph	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	1.0000
human	1.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000
interface	1.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000
intersection	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000
machine	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
management	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000
minors	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000
opinion	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000
ordered	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000
paths	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000
random	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000
response	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000
survey	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	1.0000
system	0.0000	1.0000	1.0000	2.0000	0.0000	0.0000	0.0000
testing	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000
time	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000
trees	0.0000	0.0000	0.0000	0.0000	1.0000	1.0000	0.0000
user	0.0000	1.0000	1.0000	0.0000	0.0000	0.0000	0.0000

Figure 6: Term-Document Matrix

The figure above shows the term-document matrix. Once the term document matrix is formed then the term frequency is calculated. The term frequency is nothing but the frequency of the term occurring in the document. If the document consists of five distinct terms then the term frequency of each term is 0.200. In this way the term frequency is calculated. The formula for the term frequency is given as $TF_i = N_{i,k} / \Sigma N_k$ where $N_{i,k}$ is the count of the i^{th} term in the document k and the ΣN_k is the sum of all the terms in the document k. The term frequency is calculated in this way, so as to normalize its process. After calculating the term frequency, then comes the calculation for the inverse document frequency. Inverse document frequency is necessary to determine the importance of the term. The formula for the inverse document frequency can be given as $IDF_i = \log (|D| / D_i)$ where $|D|$ is the total number of documents considered and D_i is the number of documents in which the i^{th} term is present. Once the Term Frequency TF_i and the Inverse

Document Frequency IDF_i is calculated, the $(TF\text{-}IDF)_i$ is calculated as the product of term frequency and inverse document frequency. Once the TF-IDF is calculated, and then comes the determination of similarity of the document.

Cosine similarity technique is used to determine the similarity between the two vectors in n-dimensional space. Generally to determine the similarity of the document the cosine similarity function is used. For the document similarity the vectors for the cosine similarity are generally the term frequency – inverse document frequency vectors [17]. Since the TF-IDF value of the term is always in the range of 0 to 1 the angle formed between the two vectors is always less than 90 degrees. The formula for the cosine similarity is given as COSSIM (A, B) = (A . B) / (|A| * |B|) where A and B are the two documents between which the similarity is to be determined. Once the cosine similarity is calculated then, the process of term extraction, term processing, term tokenization finishes and it is given to the clustering engine.

The algorithm for the process described above can be given as:

1. For every document read the terms one by one
2. Check if the term is stop word
    a. If it is stop word, ignore it
    b. If it is not a stop word, check if it is read
        i. If it has already been read, increase the frequency counter by 1
        ii. If it has not yet been read, add it to the list of terms with frequency count == 1
3. Form the term-document matrix

4. Calculate the term frequency
5. Calculate the Inverse Document Frequency
6. Calculate the Term Frequency – Inverse Document Frequency
7. Compute the Cosine Similarity for all the documents.

After the end of this algorithm, the process of clustering starts. The cosine similarity matrix is then given to the clustering engine.

Clustering Engine (K-EPSO algorithm):

After the computation of the document similarity, it is necessary to cluster these documents. The clustering algorithm used here is K-EPSO algorithm, which is described earlier. The clustering engine is modified to take in the cosine similarity values. The algorithm takes as input the number of documents which are considered to be the seed documents for the K-means algorithm. Then using the similarity metrics it forms the cluster of the similar documents. At the end of the K-means algorithm it forms the sufficient number of clusters which are then passed to the EPSO phase. In EPSO phase all the parameters are initialized as discussed earlier and the position of all the documents are collected. Once this is done the EPSO algorithm performs its duty and we get the clusters. The stopping criteria for the EPSO algorithm is the number of clusters formed. After the clustering process it is necessary to label those clusters. Labeling the clusters is one of the tedious steps.

Labeling of Clusters:

After the process of clustering, the clusters should be represented with the proper labels. The labeling of clusters will help the users to understand the content of the cluster.

There are many ways the labeling of clusters can be performed. One way is to use the Suffix Tree (STC), but then it becomes a long process and increases the implementation overhead. The paper [18] explains the new approach to label the clusters. Once the clusters are formed then get the documents within the cluster. From those documents count the words present in the documents which only appear in those clusters. Count all the distinct words, if the words appear more than once increment its frequency counter. After this process, sort the words in descending order. This means that the words which have the highest frequency will appear at top. Retrieve the top four words from the sorted list. These words will become the labels for the clusters which are formed earlier. The algorithm for the labeling of the clusters can be given as:

1. For every document in the cluster count the terms one by one
2. Check if the term is stop word
    a. If it is stop word, ignore it
    b. If it is not a stop word, check if it is read
        i. If it is read then increment its frequency
        ii. If it is not read, add the word to the sorting bucket and make its frequency == 1
3. Continue the process till the end of documents in the cluster
4. Sort the bucket in descending order
5. Remove the top four words to represent as labels to the cluster

After this is done the clusters are formed along with the labels. These clusters are then presented to the users through the user interface.

7. Experimental Setup:

The proposed system was built using JAVA and Eclipse IDE. Since we are doing the clustering of the web pages it is essential that the url's are presented to the user at the end of clustering process. The clustering algorithm is tested with the standard clustering data set which is discussed earlier. To get the result from the search engine, it is necessary to use the search engine API. After referring to the paper [19] which did the extensive research on the clustering engines, it came to the conclusion to use the Yahoo Search API. The other search engines provide fewer results per search. Google Search, MSN Search, Gigablast Search all retrieves 10 to 25 search results per query while the Yahoo Search retrieves 100 results per search. Also from [19] the average delay for the Yahoo API is 2.12 seconds as compared to Google API which has the average delay of 5.85 seconds. This qualifies the Yahoo API to be used in the project. The Yahoo API provides the result in the XML format. The XML parser is used to obtain the plain text documents. To map the documents perfectly the document ID is appended to it. The file is stored in this format Document ID: Plain Text of the snippet: url. The document ID is very helpful to map the plaintext data and the url's of the associated documents. This document is then passed to the algorithm for further processing. While processing of the results the Document ID is removed and also the url is removed. This helps in to only tokenize the plaintext snippet. The plaintext snippet consists of the terms necessary for the clustering. Once the clustering is done the result only consists of the Document ID. This Document ID act as the key to retrieve the url's which are discarded during the text processing step. This url's are then presented to the users as the final result.

The experiment was conducted on various search queries, to check the trueness of the output of the application. The labeling is done considering the number of similar words present in the clustered documents. The labeling is weak labeling considering the fact that the labels are generated using the number of similar words present in the document.

8. Experimental Results:

The experiment was carried out by considering various queries. Since the web pages are derived from the yahoo search, the results are not compared with any of the search engines. The results for the clustering engine were compared with the original results given in paper [11]. This is discussed in section 5 of this paper. To present the results in the paper the experiment are conducted on the ten results. The next result shows the calculation of for the raw term frequencies.

```
=== Raw Term Frequencies ===
                 D1      D2      D3      D4      D5      D6      D7      D8      D9      D10
        action  1.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
           api  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  1.0000  0.0000
  applications  0.0000  0.0000  0.0000  1.0000  0.0000  0.0000  0.0000  1.0000  0.0000  0.0000
      articles  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  1.0000  0.0000
     audiocasts  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  1.0000  0.0000
         blogs  0.0000  0.0000  0.0000  0.0000  1.0000  0.0000  0.0000  0.0000  0.0000  0.0000
         bromo  0.0000  0.0000  0.0000  0.0000  0.0000  1.0000  0.0000  0.0000  0.0000  0.0000
       capital  0.0000  0.0000  0.0000  0.0000  0.0000  1.0000  0.0000  0.0000  0.0000  0.0000
 collaboration  0.0000  0.0000  0.0000  0.0000  1.0000  0.0000  0.0000  0.0000  0.0000  0.0000
   communities  0.0000  0.0000  0.0000  0.0000  1.0000  0.0000  0.0000  0.0000  0.0000  0.0000
     community  0.0000  0.0000  1.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
      compiled  0.0000  0.0000  0.0000  1.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
      computer  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  2.0000  0.0000  0.0000  0.0000
       densely  0.0000  0.0000  0.0000  0.0000  0.0000  1.0000  0.0000  0.0000  0.0000  0.0000
      designed  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  1.0000  0.0000  0.0000
       desktop  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  1.0000  0.0000  0.0000  0.0000
       develop  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  1.0000  0.0000  0.0000
     developer  0.0000  0.0000  2.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
    developers  1.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
     different  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  1.0000  0.0000  0.0000
 documentation  0.0000  0.0000  1.0000  0.0000  0.0000  0.0000  0.0000  0.0000  1.0000  0.0000
      download  1.0000  1.0000  0.0000  0.0000  0.0000  0.0000  1.0000  0.0000  0.0000  0.0000
     downloads  0.0000  0.0000  1.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
          east  0.0000  0.0000  0.0000  0.0000  0.0000  1.0000  0.0000  0.0000  0.0000  0.0000
   environment  0.0000  1.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
    especially  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  1.0000  0.0000  0.0000
      examples  1.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
          faqs  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  1.0000  0.0000
     featuring  0.0000  0.0000  1.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
       generic  0.0000  0.0000  0.0000  1.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
```

Figure 7: Raw term frequency Results

The raw term frequency results shows the terms present in each document. The stop words appearing are deleted first and then the raw term frequency matrix is formed. After this the term frequency and the inverse document frequency is calculated. The term frequency and inverse document frequency is necessary for calculating the similarity matrix. The next shows the calculations for the cosine similarity matrix. The result shows how each document is similar to other. The figure 7 shows the similarity results for each document.

```
=== Cosine Similarity (TF/IDF) ===
           D1      D2      D3      D4      D5      D6      D7      D8      D9     D10
   D1  1.0000  0.1078  0.1194  0.0061  0.0222  0.0323  0.1002  0.0030  0.0184  0.0070
   D2  0.1078  1.0000  0.0404  0.0033  0.0022  0.0048  0.0467  0.0017  0.0018  0.0038
   D3  0.1194  0.0404  1.0000  0.0030  0.0020  0.0216  0.0422  0.0015  0.0777  0.0035
   D4  0.0061  0.0033  0.0030  1.0000  0.0021  0.0046  0.0064  0.0737  0.0017  0.0036
   D5  0.0222  0.0022  0.0020  0.0021  1.0000  0.0030  0.0186  0.0010  0.0165  0.0024
   D6  0.0323  0.0048  0.0216  0.0046  0.0030  1.0000  0.0094  0.0023  0.0025  0.0053
   D7  0.1002  0.0467  0.0422  0.0064  0.0186  0.0094  1.0000  0.0032  0.0154  0.0074
   D8  0.0030  0.0017  0.0015  0.0737  0.0010  0.0023  0.0032  1.0000  0.0009  0.0018
   D9  0.0184  0.0018  0.0777  0.0017  0.0165  0.0025  0.0154  0.0009  1.0000  0.0020
  D10  0.0070  0.0038  0.0035  0.0036  0.0024  0.0053  0.0074  0.0018  0.0020  1.0000
```

Figure 8: Cosine Similarity Results

Once the cosine similarity is done, the results are forwarded to the clustering engine where the process of clustering takes place. After this the labeling is done. And the final results with the clusters are presented to the user. The labels are generated depending upon the terms present in that document. If there are no terms present in the document then no labels are formed. The next figure shows the clustering result of the ten pages, which forms the four clusters. In this result one of the result does not have any terms associated with it, hence the cluster formed for this result does not have any of the of the

labels. This will not happen if the clusters are formed for the results which are very large and precise. The next figure shows the result.

```
=== Clusters from K-EPSO algorithm ===
java,software,computer
http://www.java.com/
http://java.sun.com/
http://www.java.net/
http://www.sun.com/java
http://java.sun.com/reference/index.jsp
https://java-net.dev.java.net/

http://www.java.com/en/download/index.jsp
java,applications,language
http://en.wikipedia.org/wiki/Java_programming_language
http://www.answers.com/topic/java
java
http://en.wikipedia.org/wiki/Java
```

Figure 9: K-EPSO Clustering Results

9. Conclusion and Future Work:

This report explains the new method of clustering. This new algorithm is used for the data clustering. The EPSO clustering technique used previously was time consuming and there was the necessity to decrease this time complexity. The new clustering algorithm which is the combination of the K-means algorithm and the Evolutionary Particle Swarm Optimization algorithm is the novel algorithm to solve the data clustering problem. To check the genuineness of this algorithm it is checked with the standard clustering data. The results obtained over there are quite satisfactory. The gain with respect to time is around 32%.

This algorithm is then used to cluster the web pages. The final result for the clustering of web pages is never tested with any other algorithm results. The main reason to do this is, other web clustering engines got different processes and then it becomes really difficult to check which one is the best algorithm.

The labeling done in this project is weak labeling, but that was the only option available to label those clusters. In future this can also be done with the Suffix Tree Clustering (STC).

This project can easily be extended further by setting the best velocity and weight for the particles considered in the EPSO phase. Also the labeling of the clusters can be improved by the new innovative way. Since much research is being conducted in the field of clustering it is pretty much sure that some other clustering techniques can be joined with this algorithm to reduce the processing time of the total clustering process.

List of References:

[1] Dubes, R. C. and Jain, A. K., Algorithms for Clustering Data, [Electronic version]. Prentice Hall (1988).

[2] Jain, A.K., M.N. Murty and P.J. Flynn, 1999. Data clustering: A review. [Electronic version]. ACM Computing Survey, 31: 264-323.

[3] Xiaohui Cui and Thomas E.Potok.: Document Clustering Analysis Based on Hybrid PSO+K-means Algorithm [Electronic version]. Journal of Computer Sciences, 2005

[4] Sara Saatchi, Chih Cheng Hung, Hybridization of Ant Colony Optimization with the K-means algorithm for clustering,[Electronic version] Lecture Notes in Computer Science St. Berlin, Germany: Springer, 2005, vol. 3540.

[5] M. Mahdavi, M. Haghir Chehreghani, H. Abolhassani, R. Forsati, "Novel meta-heuristic algorithms for clustering web documents," [Electronic version] Applied Mathematics and Computation, Science Direct, Elsevier, 2007.

[6] Khaled S. Al-Sultan, "A Tabu Search approach to a Clustering problem," [Electronic version] Pattern Recogn. 28 (1995) 1443–1451.

[7] D.W. Van der Merwe, A.P.Engelbrecht, "Data Clustering using Particle Swarm Optimization," [Electronic version] Proceedings of IEEE Congress on Evolutionary Computation 2003 (CEC 2003).

[8] J. Kennedy and R. C. Eberhart, "Particle Swarm Optimization," [Electronic version] Proceedings of IEEE International Conference on Neural Networks (ICNN), Vol. IV, Perth, Australia, 1942-1948, 1995.

[9] C. - Y. Chen, F. Ye, "Particle Swarm optimization algorithm and its application to clustering analysis", [Electronic version] Proceedings of IEEE International Conference on Networking, Sensing and Controls, 2004, pp. 789 – 794, 2004.

[10] Xiaohui Cui, Thomas E. Potok, Paul Palathingal,: "Document Clustering using Particle Swarm Optimization", [Electronic version] Proceedings of IEEE Swarm Intelligence Symposium (SIS), 2005.

[11] Shafiq Alam, Gillian Dobbie, Patricia Riddle,: "An Evolutionary Particle Swarm Optimization Algorithm for Data Clustering" [Electronic version] Proceedings of IEEE Swarm Intelligence Symposium (SIS), 2008.

[12] Tan, P., Steinbach, M., and Kumar, V.: Introduction to Data Mining, Chapter 8

[13] http://en.wikipedia.org/wiki/Application_programming_interface

[14] http://en.wikipedia.org/wiki/User_interface

[15] Manu Konchady: Text Mining Application Programming

[16] http://en.wikipedia.org/wiki/Tf-idf

[17] http://www.miislita.com/information-retrieval-tutorial/cosine-similarity-tutorial.html

[18] http://www.liaad.up.pt/~wti08/download/wti1.pdf

[19] Claudio Carpineto, Stanisiaw Osinski, Giovanni Romano, Dawid Weiss,: "A Survey of Web Clustering Engines", [Electronic version] ACM Computing Surveys, Vol 41, No 3, Article 17, July 2009.

[20] http://project.carrot2.org/

[21] http://www.seoblogr.com/google/number-of-pages-on-internet-according-to-google/

[22] http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/

[23] http://www.dbs.informatik.uni-muenchen.de/Forschung/KDD/Clustering/index.html

[24] http://www.cis.hut.fi/sami/thesis/node9.html

[25] http://en.wikipedia.org/wiki/Hierarchical_clustering