

OPTIMIZED K-MEANS CLUSTERING WITH INTELLIGENT INITIAL CENTROID SELECTION FOR WEB SEARCH USING URL AND TAG CONTENTS

S. Poomagal

Research Scholar

Department of Computer and Information Sciences

PSG College of Technology

Coimbatore, Tamilnadu, India

Mobile : 919943077744

poomagal_sam@yahoo.co.in

Dr. T. Hamsapriya

Professor and Head

Department of Information Technology

PSG College of Technology

Coimbatore, Tamilnadu, India

ABSTRACT

With the vast amount of information available online, searching results for a given query requires the user to go through many titles and snippets. This searching time can be reduced by clustering search results into clusters so that the user can select the relevant cluster at a glance by looking at the cluster labels. For web page clustering, terms (features) can be extracted from different parts of the web page. Giansalvatore, Salvatore and Alessandro have extracted terms from the entire web page for clustering. Number of terms returned in this case is more and it produces lengthy vectors. To reduce the size of the vector, Stanis law Osinski et al., and Ahmed Sameh and Amar Kadray have considered terms from the snippets.

In this work, terms are extracted from the URL (Uniform Resource Locator), Title tag and Meta tag to cluster the web documents. The reason for selecting these parts of a web page is that they have the keywords which are available in a web page. Among existing clustering algorithms, K-means algorithm is a simple algorithm and can be easily implemented for solving many practical problems. The disadvantage of K-means algorithm is the random selection of initial centroids and this paper selects initial centroids by calculating the midpoint. The proposed method of clustering is compared with Snippet based clustering and URL and Tag content based traditional K-means clustering in terms of Intra-cluster distance and Inter-cluster distance.

Categories and Subject Descriptors

H.3.3 [Information Storage and Retrieval] – Information Search and Retrieval – Clustering, Information Filtering, Retrieval Models, Search Process.

General Terms

Algorithms, Performance, Design, Experimentation, Human Factors, Standardization.

Keywords

Search Engine, Stemming, Stop words, Snippets, Clustering, Centroid, Queries, Intra-cluster distance, Inter-cluster distance, Distance function.

"Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

WIMS'11, May 25-27, 2011 Sogndal, Norway

Copyright © 2011 ACM 978-1-4503-0148-0/11/05... \$10.00"

1. INTRODUCTION

The growth of World Wide Web (WWW) is huge in the past decades. Huge volume of information is becoming available electronically over the Web. This tremendous volume poses challenges to the performance and scalability of web search engines. In addition to these technical issues, it also creates a problem for the user to browse.

In olden days, browsing a document is easy since there were only small number of documents available over the web. But nowadays, there is a freedom for anyone to create their content and host it over the web. Due to this creation, there is a difficulty in finding the necessary document. When the user gives a query, search engines return millions of documents by ordering it based on the rank of the documents. Many algorithms were used in the literature to calculate these ranks based on the popularity of a document and also its relevancy to the given query.

Even after ranking the documents using algorithms, searching is very tedious; since the number of search results displayed on each result page of a search engine is only 10 and the users have the habit of seeing only the first 10 or 20 results. It is sometimes not possible to display the relevant documents within 20 results.

A possible solution to this problem is to cluster search results into different groups that makes users to identify their required group at a glance by reading the cluster labels. Clustering is one of the most important text mining methods that help users effectively navigate, summarize and organize documents.

For search results clustering the terms were extracted from the entire web page or from the snippets by the existing methods. When selecting the contents from the entire web page[1], the clusters produced are of good quality but the size of the vector formed by this method is very large. Another difficulty with entire documents clustering is that the contents of some of the web pages could not be extracted.

When the terms are gathered only from snippets[2][3], the quality of produced clusters will not be acceptable. Since snippets are the sentences available in a web page which contain the query keywords in it. It is sometimes possible that these sentences may not have useful terms for identifying that web page.

The quality of clusters can be improved by extracting terms from the URL, Title tag and Meta tag. URL is selected since it provides the identification for the web page and the terms in it are informative. Title tag is selected since it includes the title of the page and always the title of a page has useful terms in it. Meta tag has greater significance in clustering the web pages since the major use of Meta tag on any web page is to include the keywords which are used in that page.

Once the terms are extracted from the web pages, vector space model is constructed to apply the clustering algorithm. Optimized

K-means clustering algorithm with intelligent centroid selection is used in this paper to cluster the documents.

Clustering is one of the most important text mining methods that help users effectively navigate, summarize and organize documents. By organizing large amount of documents into a number of meaningful clusters, document clustering can be used to organize the results returned by a search engine in a better way. The objective of clustering is to minimize the intra-cluster distance and to maximize the inter-cluster distance. There are many clustering algorithms available in the literature to cluster the web documents.

Suffix Tree Clustering [4] finds the base clusters by forming suffix tree with the set of phrases found in the documents. Then it combines these base clusters to form the final clusters.

Lingo clustering [2][3] clusters the documents by first generating the cluster labels and assigning documents to the clusters. For generating the labels, it finds the frequent phrases. Once the cluster labels are generated, similarity between cluster labels and input documents is identified. By comparing the obtained similarity and similarity threshold, documents are assigned to the nearest clusters.

K-means clustering clusters the documents using the traditional vector space model and then it assigns labels to each cluster by identifying the frequent phrases.

Among the existing clustering algorithms, K-means algorithm performs very well since it does not assign any document to more than one cluster. It first sets the initial parameters needed for clustering. Initial parameters are the set of centroids and the number of clusters (K). The distance from the centroids to all the documents is calculated and the documents are assigned to the nearest cluster. This process continues until the convergence criterion is met.

The problem with this algorithm is that the random selection of initial centroids and the number of clusters (K). Since initial parameters selection affects the overall performance of the algorithm, it must be selected using a defined method. This paper optimizes the performance of K-means algorithm by selecting the initial centroids in an intelligent way.

In this paper, the centroids are not selected directly from the vector space. Instead, the centroid matrix is formed with S_1 to S_{K+1} (K is the number of clusters) as rows and the terms T_1 to T_N (N is the number of terms) as columns. The entries in the centroid matrix are made by finding the scale factor from the Term Frequency / Inter-Document Frequency (TFIDF) matrix. Scale factor is identified by finding the largest number in each column of the TFIDF matrix and dividing it by the number of clusters K.

This paper focuses on clustering documents using improved K-means clustering algorithm so that each cluster contains all the web documents associated with the particular topic. Given a query and the ranked list of search results, this method first parses the web page and extracts keywords from the URL and tag contents and calculates the TFIDF values. A matrix is constructed with TFIDF values. Optimized K-means clustering with intelligent initial centroid selection is used to cluster the documents using the TFIDF matrix formed. The distance from every document to the cluster centroids is identified using Euclidean distance function.

Efficiency of the proposed method is evaluated by calculating the intra-cluster distance and inter-cluster distance. The distance between the documents within the cluster is called as intra-cluster

distance and the distance between different clusters is called as inter-cluster distance.

The remainder of this paper is organized as follows: Section 2 gives an overview of the related work. Section 3 explains the steps involved in the proposed method. Section 4 discusses the results obtained. Section 5 concludes this paper and future work is also mentioned.

2. RELATED WORK

Clustering has been considered in many different ways as alternatives to the ranked list presentation [5]. Some of them apply traditional clustering algorithms which first cluster documents into topically-coherent groups according to content similarity and then generate descriptive labels for clusters [6]. However, these labels are often unreadable, which makes it difficult for the users to identify the clusters. In this work, labels are produced by considering the terms from the title and Meta tag so that the labels are more informative and readable.

For clustering web documents, terms or phrases can be extracted from different parts of the page. A phrase is considered in [4] which is an ordered sequence of one or more words. Bigrams and trigrams are the methods used to retrieve and find meaningful phrases in natural language processing [7]. Yamamoto and Church [8] presented a method to compute all substrings term frequencies and document frequencies in large document corpora by using suffix array [9].

HuskySearch [10] retrieves results from several popular Web search engines and extracts terms from those results. Grouper [4] extracts terms from the snippets and clusters the results as they arrive using the Suffix Tree Clustering (STC) algorithm. Suffix Tree Clustering (STC) algorithm identifies the common phrases in the collection of documents. Steps in the algorithm are

- i. Base Cluster identification
- ii. Combining base clusters into final clusters

Initially, the meaningful terms are extracted by preprocessing the documents. Identification of base clusters is done by constructing the suffix tree. A suffix tree of a string S is a compact trie which contain all the suffixes of S . This algorithm considers documents as set of words, not as sequence of characters. A suffix tree is a rooted, directed tree in which each internal node has at least 2 children and each edge is labeled with a non-empty sub-string of S . The label of a node is defined to be the concatenation of the edge labels on the path from the root to that node. Each node in a suffix tree contains the document numbers in which the phrase presents. Using this, common phrases are identified. These common phrases are used to form clusters of similar documents.

Carrot2 applies Lingo clustering algorithm [2] and K-means clustering algorithm on the snippets to cluster the documents. In Lingo clustering algorithm, initially cluster labels are found and then the documents are assigned to the respective clusters. The steps in the algorithm are,

- i. Frequent phrase extraction
- ii. Cluster label induction and
- iii. Content assignment

This algorithm uses only the terms from the snippets. After extracting the meaningful terms from the snippets by performing snippet preprocessing, term-document matrix is formed and the TFIDF values are calculated and filled in the matrix. Next step of the algorithm is to find a group of cluster label candidates (Phrases or terms). It extracts frequent phrases using another algorithm namely shoc algorithm. A word-based suffix array is constructed

and extended with an auxiliary data structure called Longest Common Prefix. It takes $O(n)$ time to find all frequent phrases where n is the length of all input snippets.

During the cluster label induction phase, this algorithm identifies the abstract concepts that best describe the input collection of snippets. The steps involved in this are abstract concept discovery, phrase matching and label pruning. Once the cluster labels are produced, cluster content assignment is done using the traditional vector space model. But instead of comparing the snippets with a single query, the input snippets are matched with set of queries which are the cluster labels. Similarity among the cluster labels and the document snippet is calculated and the document is assigned to most similar cluster labels.

K-means clustering is the simplest and most commonly used clustering algorithm. Initially the algorithm takes the collection of documents, number of clusters (K) and centroids of each cluster as input. Algorithm finds the distance of documents from the initial centroids and documents are assigned to nearer clusters. This process continues until some stopping criterion is met. Selection of initial centroids and the number of clusters (K) is done randomly. Once the clusters are formed, the cluster labels are generated by finding the terms with high frequency inside the documents.

Problem with the existing clustering algorithms is that the repetition of documents in different clusters. K-means algorithm avoids this problem by assigning each document to a single cluster. Among the existing clustering algorithms, K-means clustering algorithm is the most widely used algorithm [11]. It partitions the documents into clusters by minimizing the sum of the squared distances between the documents and the centroid of the clusters.

Major drawback of K-means algorithm is that it selects the initial parameters randomly [12][13]. For improving the efficiency and accuracy of K-means algorithm, distance based algorithm is introduced [14].

Acceleration of K-means algorithm have been done by using several techniques. Kd-trees are introduced to accelerate K-means [15]. It increases the time complexity since it requires backtracking. Kd trees can be used for smaller dimensions. A new strategy for selecting the initial centroid is given by [16].

Simple Cluster Seeking [17] uses a distance value to find the centroids. It selects the first point as the cluster centroid and it continuously selects new centroid if it is d distance apart from the selected centroid. The algorithm stops when the number of centroids is equal to the number of clusters K . The major problem with this algorithm is that the selection of the centroid depends on the distance value d and there must be some defined method for selecting the distance value d .

Another work [18] concentrated on two distances $d1$ and $d2$. For each point in space, it finds the closest points with the distance less than $d1$ and it selects a new point as a centroid if it is $d2$ distance from all the selected centroids.

K-means++ [19] uses random selection of points and tests the distances of these points with the selected seed. The drawback of this random selection is that it produces different centroids for different selection of points.

SPSS (Single Pass Seed Selection) [20] provides an improvement over K-means++ by calculating the centroids in the single pass. It also uses the same distance measure as K-means++.

The problem with all the above methods of centroid selection is that it does not calculate the centroid. Instead, it selects a specific vector as a centroid. In this case, the selected centroid value may produce clusters with high intra-cluster distance and less inter-cluster distance.

3. PROPOSED METHOD

The proposed method comprises the following phases:

1. Term Extraction (URL, Title tag and Meta tag)
2. Stop word removal and Stemming
3. TFIDF calculation
4. Applying clustering algorithm
5. Cluster label generation

Pseudo-code of the proposed method

```

1:  $D \leftarrow$  Documents (URL & Tag contents)
   {STEP 1: Preprocessing}
2: For all  $d \in D$  do
3:   Compare each term with the stop words list
4:   If any match found then
5:     Remove the matched terms;
6:   End if
7:   Apply Stemming Algorithm
8: End for
   {STEP 2: Duplicates Elimination}
9: Concatenate all the extracted terms into a single
   document  $d$ ;
10: For all the terms in  $d$ 
11:   Find the distinct terms  $t_i$ ;
12: End for
   {STEP 3: TFIDF Calculation}
13: For all  $t_i \in d$ 
14:   For all  $j \in D$ 
15:      $TFIDF_{ij} = TF_{ij} * IDF_i$ 
16:   End for
17: End for
   {STEP 4: Applying Improved K-means}
18: Apply Improved K-means to find the Clusters
   {STEP 5: Cluster Label Creation}
19: For all  $C_i \in C$  do
20:   Extract the terms from the title and meta tags of
   the documents in  $C_i$ 
21:   Find the frequencies of all the terms
22:   Select the term with high frequency value as the
   cluster label
23: End for

```

Initially, the search results returned by a web search engine are retrieved. These web pages are analyzed by a parser and the contents are extracted from the URL and the specified tags. These contents are informative enough because most search engines are well designed to facilitate users' relevance judgement only by the tag contents. Stop words are removed from the collection of words by checking it against the database of stop words and stemming is then performed to convert the words into their root form. In this paper, Porter's algorithm is used for stemming.

TFIDF matrix is formed by measuring the Term Frequency and Inter-Document Frequency values. Clustering algorithm is applied on the matrix formed. In this paper, optimized K-means clustering algorithm is used to cluster the documents and Euclidean distance

function is used to calculate the distance between the documents and the cluster centroids.

In traditional K-means clustering algorithm, cluster centroids are selected randomly. In improved K-means algorithm, initial cluster centroids are selected by calculating the scale factor. Once the initial centroids are formed, each document is assigned to the cluster for which the document is having the minimum distance. After forming clusters, new centroids are calculated for each cluster. New centroids are calculated by taking the average of the term frequencies of the documents inside the clusters. This process is continued until the results of the two iterations become the same.

3.1 Term Extraction

Text document clustering is different from web document clustering. Web documents are unstructured. In addition to text contents it contains tag information. This tag information and the punctuations should be removed from the document to extract meaningful terms. Tokenization is used to perform this operation.

Given a document, tokenization is the task of breaking into pieces, called tokens, perhaps at the same time throwing away certain characters, such as punctuation. Here is an example of tokenization:

Input: Friends, Romans, Country men, lend
me your ears;

Output: Friends Romans Country men lend
me your ears

These tokens are often loosely referred to as terms or words, but it is sometimes important to make a distinction between a type and a token. A token is an instance of a sequence of characters in some particular document that are grouped together as a useful semantic unit for processing. A type is the class of all tokens containing the same character sequence. A term is a type that is included in the Information Retrieval (IR) system's dictionary.

Contents from the URL and the specified tags are extracted by removing the tags and special symbols with the use of tokenization. Since the number of terms extracted is less, there is no need to perform feature selection. Instead, all the extracted terms can be considered as features for clustering the web documents.

3.2 Stop word removal and stemming

Stop words does not provide any useful information to perform clustering. So they can be removed to avoid confusions. Some common stop words are is, was, are, were, what etc., For stop word removal, first the database of stop words are created and the terms extracted from the web pages are compared with the stop words list. Stop words found in the extracted terms are removed.

The goal of stemming is to reduce inflectional forms and sometimes derivationally related forms of a word to a common base form. For instance:

am, are, is $\Rightarrow$ be
car, cars, car's, cars' $\Rightarrow$ car
the boy's cars are different colors $\Rightarrow$
the boy car be differ color

Stemming usually refers to a crude heuristic process that chops off the ends of words and often includes the removal of derivational affixes.

In many 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 stemming algorithms or stemmers have been developed, that attempt to reduce a word to its base form or root form. Stemming reduces the size of the collection of words, that is, the number of distinct terms needed for representing a set of documents. This smaller size results in a saving of storage space and processing time.

Usually the stemmed words are not meaningful. For example, the stemmed word of "computation" is "comput". While stemming the words, two things have to be considered.

- i. Different words with the same base meaning are converted to the same form
and
- ii. Words with distinct meanings are kept separate.

For stemming, Porter's algorithm is used in this paper. It is a simple utility that reduces English words to their word stems - without the "ing", "ings", "s" etc.,

Table 1 shows the sample stemmed words:

Word	Base Word	Word	Base Word
Consigned	Consign	Consolation	Consol
Consigning	Consign	Consolations	Consol
Consignment	Consign	Console	Consol
Consisted	Consist	Consoled	Consol
Consistency	Consist	Consoles	Consol
Consistently	Consist	Consonant	Conson
Consisting	Consist	Consorted	Consort
Consists	Consist	Conspirator	Conspir

Table 1 : Words and their equivalent base words

3.3 TFIDF Calculation

TFIDF produce a composite weight for each term in each document. This assigns to term i a weight in document j given by

$$TFIDF_{i,j} = TF_{i,j} * IDF_i$$

TF_{i,j} is calculated as:

$$TF_{i,j} = N_{i,j} / NT_j$$

N_{i,j} is the number of times the term i appears in the document j and NT_j is the total number of terms in the document j.

The inverse document frequency (IDF_i) is calculated as:

$$IDF_i = \log(|D| / |d : t_i \in d|)$$

Where |D| is the total number of documents and |d : t_i ∈ d| is the number of documents in which the term t_i appears. These term frequencies and the list of documents are then formed as a vector space.

3.4 K-means and Improved K-means Clustering

K-means clustering is the simplest and most commonly used clustering algorithm. Initially the algorithm takes the collection of documents, number of clusters (K) and centroids of each cluster as input. Algorithm finds the distance of documents from the initial centroids and documents are assigned to nearer clusters. This process continues until some stopping criteria is met. Selection of initial centroids is done randomly.

Steps in K-means Clustering algorithm

1. Choose initial cluster centroids to coincide with K randomly selected documents from the document set.
2. Assign each document to its closest cluster.

3. Recompute the cluster centroids using the current cluster memberships.
4. If there is a reassignment of documents to the new cluster, go to step 2. Typical stopping criteria is : Groups formed by the subsequent iterations must be same.

Pseudo code for traditional K-means

- 1: Randomly select K cluster centroids (C);
- 2: Repeat
- 3: For all $c \in C$ do
- 4: Find the distance between the document and the cluster centroid using the following formula

$$\sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$
- 6: Fill the corresponding distance in the distance table
- 7: End for
- 8: Until $\text{table}_j = \text{table}_{j-1}$

In this method, cluster centroids are chosen randomly. Selection of centroid decides the convergence time of the algorithm. For the same data set, different centroid selection will produce different kinds of clusters. This can be avoided by selecting the initial centroids using the following strategy.

Steps in the Improved K-means Algorithm

1. Assign the initial value of K (No of clusters to be created) as $\sqrt[3]{n/2}$.
2. Select K initial centroids by using the proposed method
3. Assign each document to the nearest cluster
4. Recompute the cluster centroids using the current cluster memberships.
5. If there is a reassignment of documents to the new cluster, go to step 3. A typical stopping criterion is : Groups formed by subsequent iterations must be same.

Centroid selection proceeds in the following manner. Consider a TFIDF matrix in which columns belongs to terms and the rows belongs to documents. On each column, find the largest value and divide that value by the number of clusters (K). Use this value as a scale factor and form another matrix with the terms T_1 to T_N as columns and S_1 to S_{k+1} as rows. Enter values in each column of the formed matrix starting from 0 and increment the value using the scale factor of that term computed previously. Calculate the mean of the values in the adjacent rows and the calculated mean vector is the new centroid vector.

Steps in the selection of initial Centroids

1. Calculate the scale factor (SF) from the TFIDF matrix

$$SF = \text{MAX}(T_i) / K$$

Where T_i is the i^{th} term in the TFIDF matrix and i changes from 1 to the number of terms in the TFIDF matrix and K is the Number of Clusters

2. Form another matrix with T_1 to T_N as columns and S_1 to S_{k+1} as rows and fill in each column starting from 0 and increase the value using the scale factor.
3. Calculate the centroids by taking the mean of the values in adjacent rows of the newly created matrix.

Example

- Consider the following TFIDF matrix,

	T1	T2	T3	T4	T5
D1	4	7	9	20	12
D2	6	10	25	10	15
D3	3	14	36	13	18
D4	8	9	10	12	14

- Assume the value of K to be 2
- Scale factor calculated for each term is,

$$\begin{aligned} T1 &= \text{MAX}(T1) / 2 = 8/2 = 4 \\ T2 &= \text{MAX}(T2) / 2 = 14/2 = 7 \\ T3 &= \text{MAX}(T3) / 2 = 36/2 = 18 \\ T4 &= \text{MAX}(T4) / 2 = 20/2 = 10 \\ T5 &= \text{MAX}(T5) / 2 = 18/2 = 9 \end{aligned}$$

- Since the number of clusters is initialized as 2, new matrix formed has 3 rows (K+1) as follows:

	T1	T2	T3	T4	T5
S1	0	0	0	0	0
S2	4	7	18	10	9
S3	8	14	36	20	18

- Now the centroids are calculated and the centroids are,

$$\begin{aligned} C1 &= \{(0+4)/2, (0+7)/2, (0+18)/2, (0+10)/2, (0+9)/2\} \\ &= \{2, 3.5, 9, 5, 4.5\} \\ C2 &= \{(4+8)/2, (7+14)/2, (18+36)/2, (10+20)/2, (9+18)/2\} \\ &= \{6, 10.5, 27, 15, 13.5\} \end{aligned}$$

- C1 and C2 are the initial centroids.

Pseudo code for the selection of initial centroids

- 1: K ← Number of clusters
- M ← TFIDF matrix
- {STEP 1: Finding maximum in each row}
- 2: X=0
- 3: For i=1 to r do
- 4: max=0;
- 5: For j=1 to c do
- 6: if $M[i][j] > \text{max}$
- 7: max=M[i][j]
- 8: C[X]=max
- 9: X=X+1
- 10: End if
- 11: End for
- 12: End for
- {STEP 2: Scale factor identification}
- 13: For i=1 to X do
- 14: C[i]=C[i]/K
- 15: End for
- {STEP 3: Centroid matrix}
- 16: For i=1 to r do
- 17: Y=0
- 18: For j=1 to K+1 do
- 19: cent[i][j]=Y
- 20: Y=Y+c[i]
- 21: End for
- 22: End for
- {STEP 4: Forming Centroid Vectors}
- 23: For i=1 to K do
- 24: For j=1 to r do

```

25:   centvect[i][j]=abs((cent[i][j]-cent[i+1][j])/2
26:   End for
27: End for
28: Repeat
29:   For all centvecti ∈ C do
30:     Find the distance between the document
       and the cluster centroid using the
       following formula

```

$$\sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

```

31:   Fill the corresponding distance in the
       distance table
32:   End for
33: Until tablej==tablej-1

```

3.5 Cluster Label Generation

Cluster labels are generated to provide identification for the clusters. Creation of labels is done by extracting the terms from the documents in each cluster. Since the extraction of terms from the entire document may produce more number of terms, only Title and Meta tag contents are extracted. For all the extracted terms, term frequency is calculated and the term which is having highest term frequency is taken as the label for the cluster.

4. Experimental Results

For the experiment, 200 queries are considered. For all the 200 queries, first 200 results from yahoo, google and bing are retrieved. Code is written in PHP and the collected web documents are stored using SQL Server. The measures such as intra-cluster distance and inter-cluster distance are taken. Existing snippet based clustering algorithm, URL and Tag content based traditional K-means clustering algorithm and URL and Tag content based optimized K-means clustering algorithm are compared in terms of these measures.

First the terms are extracted from URL and tag contents and the stop words are removed. Table 2 shows the list of stop words considered in this paper.

a	a, able, about, above, according, accordingly, across, actually, after, afterwards, again, against, ain't, all, allow, allows, almost, alone, along, already, also, although, always, am, among, amongst, an, and, another, any, anybody, anyhow, anyone, anything, anyway, anyways, anywhere, apart, appear, appreciate, appropriate, are, aren't, around, as, aside, ask, asking, associated, at, available, away, awfully
b	be, became, because, become, becomes, becoming, been, before, beforehand, behind, being, believe, below, beside, besides, best, better, between, beyond, both, brief, but, by
c	common, came, can, can't, cannot, cant, cause, causes, certain, certainly, changes, clearly, co, com, come, comes, concerning, consequently, consider, considering, contain, containing, contains, corresponding, could, couldn't, course, currently
d	definitely, described, despite, did, didn't, different, do, does, doesn't, doing, don't, done, down, downwards, during,
e	each, edu, eg, eight, either, else, elsewhere, enough, entirely, especially, et, etc, even, ever, every, everybody, everyone, everything, everywhere, ex, exactly, example, except
f	far, few, fifth, first, five, followed, following, follows, for, former, formerly, forth, four, from, further, furthermore
g	get, gets, getting, given, gives, go, goes, going, gone, got, gotten, greetings
h	had, hadn't, happens, hardly, has, hasn't, have, haven't, having, he, he's, hello, help, hence, her, here, here's, hereafter,

	hereby, herein, hereupon, hers, herself, hi, him, himself, his, hither, hopefully, how, howbeit, however
i	i'd, i'll, i'm, i've, ie, if, ignored, immediate, in, inasmuch, inc, indeed, indicate, indicated, indicates, inner, insofar, instead, into, inward, is, isn't, it, it'd, it'll, it's, its, itself
j	Just
k	keep, keeps, kept, know, knows, known
l	last, lately, later, latter, latterly, least, less, lest, let, let's, like, liked, likely, little, look, looking, looks, ltd
m	mainly, many, may, maybe, me, mean, meanwhile, merely, might, more, moreover, most, mostly, much, must, my, myself
n	name, namely, nd, near, nearly, necessary, need, needs, neither, never, nevertheless, new, next, nine, no, nobody, none, nor, normally, not, nothing, novel, now, nowhere
o	obviously, of, off, often, oh, ok, okay, old, on, once, one, ones, only, onto, or, other, others, otherwise, ought, our, ours, ourselves, out, outside, over, overall, own
p	particular, particularly, per, perhaps, placed, please, plus, possible, presumably, probably, provides
q	que, quite, qv
r	rather, rd, re, really, reasonably, regarding, regardless, regards, relatively, respectively, right
s	said, same, saw, say, saying, says, second, secondly, see, seeing, seem, seemed, seeming, seems, seen, self, selves, sensible, sent, serious, seriously, seven, several, shall, she, should, shouldn't, since, six, so, some, somebody, somehow, someone, something, sometime, sometimes, somewhat, somewhere, soon, sorry, specified, specify, specifying, still, sub, such, sup, sure
t	t's, take, taken, tell, tends, th, than, thank, thanks, thanx, that, that's, thats, the, their, theirs, them, themselves, then, thence, there, there's, thereafter, thereby, therefore, therein, theres, thereupon, these, they, they'd, they'll, they're, they've, think, third, this, thorough, thoroughly, those, though, three, through, throughout, thru, thus, to, together, too, took, toward, towards, tried, tries, truly, try, trying, twice, two
u	un, under, unfortunately, unless, unlikely, until, unto, up, upon, us, use, used, useful, uses, using, usually
v	value, various, very, via, viz, vs
w	want, wants, was, wasn't, way, we, we'd, we'll, we're, we've, welcome, well, went, were, weren't, what, what's, whatever, when, whence, whenever, where, where's, whereafter, whereas, whereby, wherein, whereupon, wherever, whether, which, while, whither, who, who's, whoever, whole, whom, whose, why, will, willing, wish, with, within, without, won't, wonder, would, would, wouldn't
y	yes, yet, you, you'd, you'll, you're, you've, your, yours, yourself, yourselves
z	Zero

Table 2: List of Stop words

Since only the contents of URL and specific tags are selected, number of words after stop word removal is lesser than the number of words produced by selecting entire document. Table 3 shows the number of words before stop word removal and after stop word removal.

KEYWORDS / NUMBER OF DOCUMENTS	NUMBER OF WORDS EXTRACTED			NUMBER OF WORDS AFTER STOPWORD REMOVAL		
	50	100	200	50	100	200
APPLE	905	1787	2626	756	1338	1997
MYSQL	452	854	1565	377	688	1233

FREE ANTIVIRUS	492	763	1291	408	613	1024
BRIDGE	470	921	1951	397	790	1670
CLUSTER	803	1233	2272	679	1017	1780
DICTIONARY	113	1469	2229	982	1232	1833
PROCESSOR	483	979	2257	407	809	1760
NIMCET	532	819	1451	423	642	1142
MOUSE	921	1515	2445	766	1237	1973
TAMIL MP3	948	1407	2137	843	1234	1847
MOBILE	630	875	1392	461	646	1088
CAMERA	457	847	1828	358	673	1448
HARDWARE	397	684	1303	346	578	1701
GRID COMPUTING	433	653	1394	344	513	1106
NETWORKING	581	929	1780	487	767	1404
ORANGE	653	1490	3019	566	1266	2434
DATASTRUCTURES	363	763	1349	288	592	1028
DOCOMO	553	1089	1940	460	879	1525
CLOUD COMPUTING	381	613	1145	312	516	927
CAT	645	1298	2274	533	1063	1865

Table 3: Stop word removal

Once the stopwords are removed from the collection of extracted terms, stemming is performed to convert the terms into their base form. Table 4 shows the stemmed terms produced.

redplaid, cloud, oracl, weblogic, comput, easy, google, chelsio, communications, host, free, trial, wikipedia, alpha, wmf, definit, whati, word, http, sdefinit, sid, gci, html, noodp, concept, articl, describ, impact, variety, compani, countri, industri, refer, amazon, amzn, blackbaud, blk, blackboard, bbbb, concur, technolog, cnqr, geographi, page, googl, goog, hp, intel, intc, intern, busi, machin, ibm, new, result, salesforc, forc, platform, leader, servic, instant, time, valu, complet, independ, noydir, really, mean, infoworldth, big, trend, sound, nebul, fuzzy, view, proposit, perspect, profession, index, follow, web, app, engin, applic, architectur, infrastructur, soa, orient, softwar, sun, microsystem, open, sourc, discov, brought, world, popular, starlight, howstuffwork, quot, work, smart, essenti, futur, let, file, internet, learn, benefit, drawback, computingbarry, warwick, raleigh, ibmu, public, v, templat, gener, corpor, copyright, expo, sy, media, journalsy, computingcloud, n, june, stamford, ct, youtub, explain, computingexplain, video, produc, present, christoph, barnatt, author, explainingcomput, associ, professor, compu, utility, saa, haa, explainedcloud, confus, term, want, talk, technology, buzzword, boil, section, ec, gogrid, netflix, gmail, server, dedic, vmware, virtualizationturn, datacent, secur, privat, virtual, add, capacity, demand, deploy, extern, built, vm, ware, gklqyx, kxtfwei, optim, data, center, lab, consolid, lifecycl, provid, continuity, pc, solut, dell, increas, capabl, flexibility, choic, eiwatch, content, enterpris, aspx, ng, publish, elast, ecamazon, aw, deliv, set, form, reliabl, scalabl, inexpens, pay, includ, simpledb, sq, fp, simpl, storag, grid, mechan, turk, queue, flexibl, payment, devpay, alexa, develop, forum, user, group, tool, api, unveil, centr, india, nist, gov, security, divis, csrc, websit, involv, differ, project, webpage, base, visit, resourc, clearinghous, event, driver, mobileag, pki, encrypt, ae, mode, ipv, alert, virus, organ, link, mission, staff, itl, division,

network, policy, manag, oper, solutions, ubuntu, plain, english, common, craft, product, computerworld, knowledge, voic, inform, c, right, reserv, action, command, viewpag, amp, pagepath, polici, eng, document, faq, demystifi, worldnetwork, bring, great, promis, industry, key, question, answer, level, agreement, supplement, jon, brodkin, schneier, onlin, rackspac, mosso, power, site, support, live, chat, today

Table 4 : Stemmed terms

Quality of the produced clusters is identified using intra-cluster distance and inter-cluster distance.

Intra-cluster distance is calculated by finding square of the summation of the absolute differences between each point in a cluster and adding a constant with it and finally dividing the result by 2.

$$L1 = \frac{1}{2} \left(\sum_{k \in C_k} |x_i - x_j|^2 + \text{constant} \right)$$

Inter-cluster distance is calculated by finding the square of distance between every point with all the other points in other clusters and adding a constant with it and finally dividing it by 2.

$$L2 = \frac{1}{2} \left(\sum_{\substack{i \in C_k \\ j \notin C_k}} |x_i - x_j|^2 + \text{constant} \right)$$

Figure 1 shows the intra-cluster distances of clusters produced by Snippet based clustering, URL and Tag content based traditional K-means clustering and URL and Tag content based optimized K-means clustering with intelligent initial centroid selection.

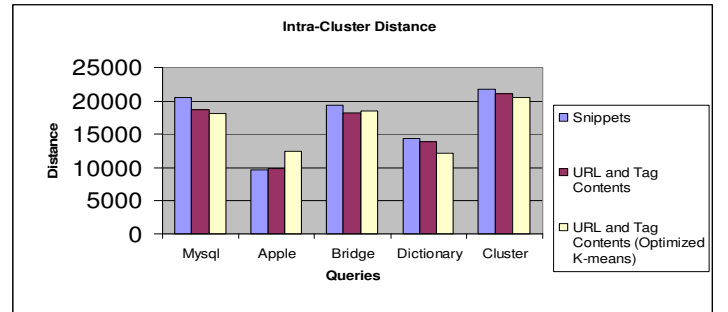


Figure 1: Intra-cluster Distance

The above figure shows that the clusters produced by the proposed method have less intra-cluster distance so that the documents in each cluster are highly related to one another.

Figure 2 shows the inter-cluster distances of clusters produced by Snippet based clustering, URL and Tag content based traditional K-means clustering and URL and Tag content based optimized K-means clustering with intelligent initial centroid selection.

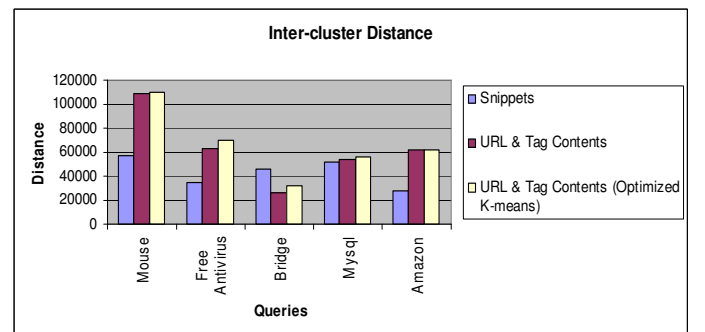


Figure 2: Inter-cluster Distance

The above figure proves that the proposed method have produced clusters such that the documents in each cluster are highly independent of the documents in other clusters.

5. CONCLUSION

In this paper, terms are extracted from URL, Title tag and Meta tag so that the vectors produced by this method are not too long when compared to entire document contents. Also this method produced high quality clusters than snippet based clustering since the terms in URL and specified tags are having greater significance in providing identification for the page. Among the existing clustering algorithms, K-means algorithm is the best algorithm for clustering larger data sets. The major drawback of this algorithm is the random selection of initial parameters such as centroid selection and the number of clusters selection. In this work, K-means clustering is optimized by selecting the initial centroids in an intelligent way. The results show that the proposed method of clustering search results has produced high quality clusters where the documents inside each cluster are tightly coupled. In future, this work can be extended by introducing a new method for the selection of the number of clusters (K) and also the terms can be extracted from other parts of the web page.

REFERENCES

- [1] Giansalvatore Mecca, Salvatore Raunich and Alessandro Pappalardo (2007) "A new algorithm for clustering search results", *ACM Transactions on Data & Knowledge Engineering*, vol. 62, no. 3, pp. 504-522.
- [2] Stanislaw Osinski, Jerzy Stefanowski and Dawid Weiss (2004) "Lingo: Search Results Clustering Algorithm Based on Singular Value Decomposition". *Advances in Soft Computing, Intelligent Information Processing and Web Mining, Proceedings of the International IIS: IIPWM'04 Conference, Zakopane, Poland*, pp. 359—368.
- [3] Ahmed Sameh. and Amar Kadray (2010), 'Semantic Web Search Results Clustering Using Lingo and WordNet', *International Journal of Research and Reviews in Computer Science (IJRRCS)*, Vol. 1, No. 2, pp.71-76.
- [4] O. Zamir, O. Etzioni (1999) "Grouper - A Dynamic Clustering Interface to Web Search Results", *Computer Networks*, vol. 31, no. 11-16, pp. 1361-1374.
- [5] A. Leouski and J. Allan (2000) "Improving Interactive Retrieval by Combining Ranked List and Clustering", *Proceedings of RIAO*, pp. 665-681.
- [6] Dmitri V Kalashnikov, Zhaoqi (Stella) Chen, Sharad Mehrotra and Rabia Nuray-Turan (2008) "Web People Search via Connection Analysis", *IEEE Transactions on Knowledge and Data Engineering*, vol.20, no.11, pp. 1441-1457.
- [7] E. Charniak, (1993) "Statistical Language Learning", MIT Press, 1993.
- [8] M. Yamamoto and K. W. Church (2001) "Using Suffix Arrays to Compute Term Frequency and Document Frequency for all Substrings in a Corpus", *Computational Linguistics*, vol. 22, no. 1, pp. 1-30.
- [9] U. Manber and G. Myers (1993) "Suffix Arrays: A New Method for On-Line String Searches", *SIAM Journal of Computing*, vol. 22, no.5, pp. 935-948.
- [10] E. Selberg and O. Etzioni (1995) "Multi-service search and comparison using the MetaCrawler", *Proceedings of the 4th World Wide Web Conference (WWW4)*, pp. 195-208.
- [11] J. MacQueen (1967) "Some methods for classification and analysis of multivariate observations", *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probabilities*, pp:281-297, 1967.
- [12] T. Kanungo, D. M. Mount, N. Netanyahu, C. Piatko, R. Silverman and A. Y. Wu (2002) "An efficient K-means clustering algorithm: Analysis and implementation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24 (7): 881-892.
- [13] D. Pelleg and A. Moore (1999) "Accelerating exact K-means algorithms with geometric reasoning", In *Proceedings of the Conference on Knowledge Discovery in Databases (KDD 99)*, San Diego, California, pp. 277–281.
- [14] K. A. Abdul Nazeer and M. P. Sebastian (2009) "Improving the accuracy and efficiency of the K-means clustering algorithm", *Proceedings of the world congress on Engineering, WCE 2009, London, UK*.
- [15] J. Bentley (1995) "Multidimensional Binary Search Trees Used for Associative Searching", *Communications. ACM*, vol. 18, no. 9, pp. 509-517.
- [16] Zhang, Chen and Xia Shixiong (2009) "K-means Clustering Algorithm with improved initial Center", *Second International Workshop on Knowledge Discovery and Data Mining*, pp.23-25.
- [17] J. Tou and R. Gonzales (1997) "Pattern Recognition Principles", Addison-Wesley, Reading, MA., ISBN : 0201075873.
- [18] M. M. Astrahan (1970) "Speech analysis by clustering or the hyperphoneme method", Stanford University, California, Accession No : AD0709067.
- [19] D. Arthu and Vassilvitskii (2007) "K-means++: The advantages of careful seeding", *Proceeding of the 18th Annual ACM-SIAM Symposium of Discrete Analysis*, January 7-9, ACM Press, New Orleans, Louisiana, pp: 1027 – 1035.
- [20] K. Karteeka Pavan, A. V. Allam Appa Rao, Datatreya Rao and G. R. Sridhar (2010) "Single Pass Seed Selection Algorithm for K-means", *Journal of Computer Science* vol.6, no. 1, pp. 60-66.