Annals of Pure and Applied Mathematics
Vol. 5, No.1, 2013, 47-52
ISSN: 2279-087X (P), 2279-0888(online)
Published on 13 November 2013
www.researchmathsci.org

Annals of
Pure and Applied
Mathematics

Design and Analysis of Algorithm to Search Documents with Misspelled Input

***S.Mandal*[1] and *A.Pal*[2]**

[1]Department of Computer Science and Application,
Dr Harisingh Gour Central University, Sagar,
Madhya Pradesh-470003, India
Email: shrabmandal@gmail.com
[2]Department of Mathematics, National Institute of Technology Durgapur,
Durgapur-713209, West Bengal, India
Email: anita.buie@gmail.com

Received 28 September 2013; accepted 5 October2013

Abstract. Clustering is the useful technique for discovering of data distribution and patterns in the underlying data. The goal of clustering is to discover both the dense and the sparse regions in a data set. The main emphasis has been to cluster with high accuracy as possible and search the required data from them, while keeping the input/output (I/O) cost high. Thus, it is necessary to investigate the principle of clustering and searching to design efficient algorithms which meet the specific requirement of minimizing the I/O operation. So our objective is to build a searching tool by using the Meta-Clustering Technique which means combination of more than one concept for clustering the data.

Our experimental result says if a user gives a wrong spelling then at least one output will be common if he will give the correct spell. Our proposed searching tool works in the static database but in future it will be able to work in the dynamic database even may be used as web search engine.

Keywords: Clustering, K-Means Algorithm, Searching tool, Meta-Clustering

1. Introduction

The typical information retrieval model in current searching tools is to first retrieve the relevant documents based on the user input query, i.e. the name of file. Some searching tools are able to retrieve the documents by the content. In our work we have tried to retrieve data even when the user entered the misspelled keyword. This paper is designed in five sections. First section describes the clustering technique and applying it. Second section contains the idea of correcting error of entered keyword. Third section describes the searching. Next section contains the analysis of result and the last section deals with the conclusions and future scope.

2. Clustering Technique

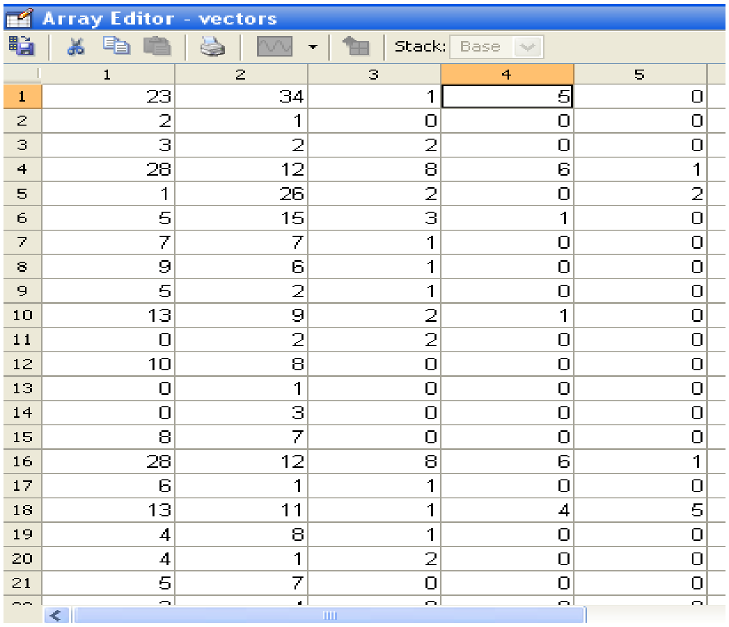
Over the recent past organizations and other users have been capturing increasingly large amounts of data that they wish to analyze. The amount of data being collected in databases today far exceeds the ability to reduce and analyze data without the use of automated analysis techniques. Knowledge Discovery in Databases (KDD) is an interdisciplinary field that is evolving to provide automated analysis solutions. The core part of the KDD process is the application of specific data mining methods for pattern discovery and extraction. Among the various data mining techniques, clustering of data plays a major role in extracting knowledge from the existing database. In this paper we focus on the k-means clustering technique and some other concepts, called Meta-clustering Technique.

K-MEANS Algorithm

Our technique is based on the k-means clustering method. The working principle of it is described below:

(1) Arbitrarily choose k objects as the initial cluster centers.
(2) Repeat step 3 to step 5
(3) Reassign each object to the cluster to which the object is the most similar based on the mean value of the objects in the clusters.
(4) Update the cluster means .i.e., calculate the mean value of the object's for each cluster.
(5) Until no change.

For applying any clustering algorithm, input is coordinates of the data files. The coordinates of the file is decided by the word which appeared more frequently than others. So we may not identify the coordinates of the files .This is the one of the important task of our technique to calculate the correct coordinate. In our paper coordinates are obtained by vectorization technique between files in database and input categorical files. The coordinates of the files are shown in table 1.1.



	1	2	3	4	5
1	23	34	1	5	0
2	2	1	0	0	0
3	3	2	2	0	0
4	28	12	8	6	1
5	1	26	2	0	2
6	5	15	3	1	0
7	7	7	1	0	0
8	9	6	1	0	0
9	5	2	1	0	0
10	13	9	2	1	0
11	0	2	2	0	0
12	10	8	0	0	0
13	0	1	0	0	0
14	0	3	0	0	0
15	8	7	0	0	0
16	28	12	8	6	1
17	6	1	1	0	0
18	13	11	1	4	5
19	4	8	1	0	0
20	4	1	2	0	0
21	5	7	0	0	0

Table 1.1: Representing the vectors

The k-means clustering technique is used here for clustering the database. To cluster the database we use some categorical data files because we want to represent the data within cluster as hierarchical order .This gives the fast search time.

We also observe the size of each cluster because large cluster does not give the efficient search time.

So whenever it generates a large one this will be clustered again. Steps in this module are given below:

Step 1: Prepare database and categorical data files.
Step 2: Repeat step 3 to step 10 for each file.
Step 3: Repeat step 4 to step 5
Step 4: Perform dot (.) operation between file and all categorical data files.
Step 5: Output of dot (.) operation is a coordinate stored in a matrix called vector.
Step 6: Apply k-means algorithm to vector matrix.
Step 7: Newly generated cluster's size is checked.
Step 8: If cluster size is large then the one fourth of vector matrix goes to step 5.
Step 9: If a cluster contains a less amount of data then merge it with another cluster which has also less amount of data.
Step 10: Finally store the final clusters in a final_cluster file and centroid of clusters in centroid_cluster matrix.

After applying the repeatedly k-means and merging the clusters we get the final cluster which is given in figure 1.The centroid of the clusters are calculated and represented in table 1.2.

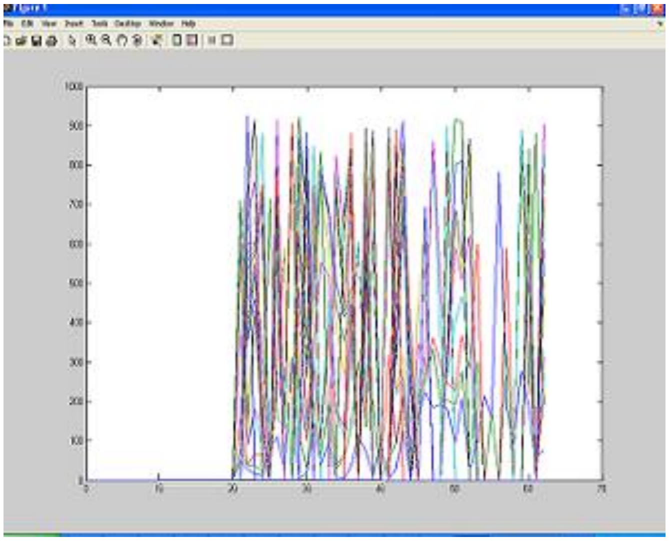
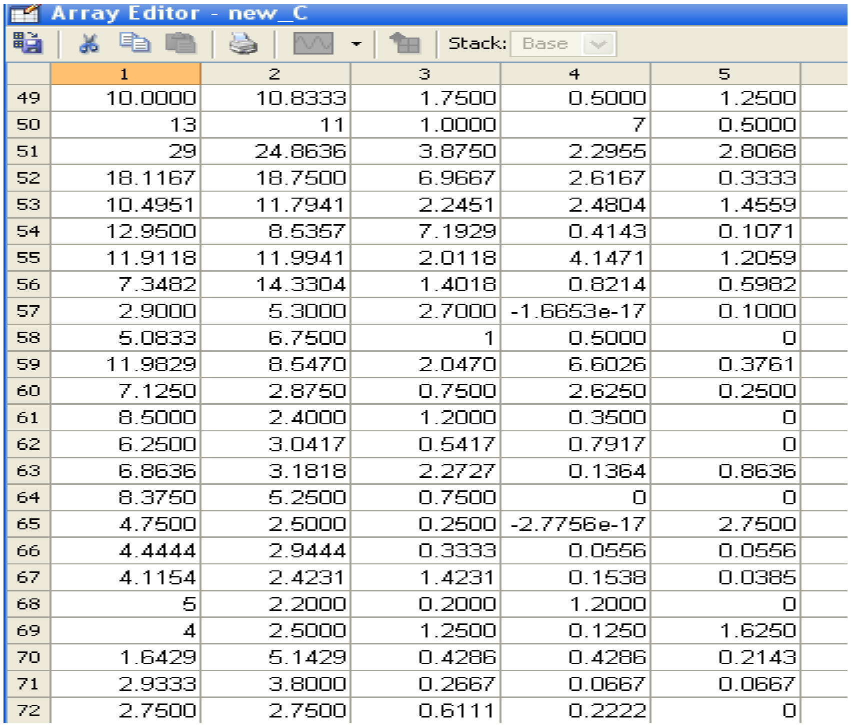


Figure 1: Representing the final_cluster



	1	2	3	4	5
49	10.0000	10.8333	1.7500	0.5000	1.2500
50	13	11	1.0000	7	0.5000
51	29	24.8636	3.8750	2.2955	2.8068
52	18.1167	18.7500	6.9667	2.6167	0.3333
53	10.4951	11.7941	2.2451	2.4804	1.4559
54	12.9500	8.5357	7.1929	0.4143	0.1071
55	11.9118	11.9941	2.0118	4.1471	1.2059
56	7.3482	14.3304	1.4018	0.8214	0.5982
57	2.9000	5.3000	2.7000	-1.6653e-17	0.1000
58	5.0833	6.7500	1	0.5000	0
59	11.9829	8.5470	2.0470	6.6026	0.3761
60	7.1250	2.8750	0.7500	2.6250	0.2500
61	8.5000	2.4000	1.2000	0.3500	0
62	6.2500	3.0417	0.5417	0.7917	0
63	6.8636	3.1818	2.2727	0.1364	0.8636
64	8.3750	5.2500	0.7500	0	0
65	4.7500	2.5000	0.2500	-2.7756e-17	2.7500
66	4.4444	2.9444	0.3333	0.0556	0.0556
67	4.1154	2.4231	1.4231	0.1538	0.0385
68	5	2.2000	0.2000	1.2000	0
69	4	2.5000	1.2500	0.1250	1.6250
70	1.6429	5.1429	0.4286	0.4286	0.2143
71	2.9333	3.8000	0.2667	0.0667	0.0667
72	2.7500	2.7500	0.6111	0.2222	0

Table 1.2: Centroid_cluster

3. Text Correction

This module helps us to correct spelling of keywords. For correcting the spelling it generates the possible combination of input keyword. In the next module we will search all the possible keywords from the files which are stored in database.

To generate the all possible combination of the input keyword we use a matrix. This matrix has 26 rows and 3 columns. Every row contains that three alphabets which are similar sounding words and uses make frequently typing mistake because of similar looks and appearance in nearest distance in keyboard.

Here we input the similar matrix which is given bellow:

$$\text{Simi1} = \begin{bmatrix} a & e & o \\ b & v & b \\ c & k & e \\ d & t & f \\ . & . & . \\ . & . & . \\ w & v & q \\ x & a & e \\ y & u & v \\ z & j & x \end{bmatrix}$$

Figure 2: Simi1 matrix

The steps which are followed in text correction module are given below

Step 1: Accept the user keyword.
Step 2: Perform the dot (.) operation between each row of simil matrix and input keyword.
Step 3: The output of dot (.) operation is the coordinate of input keyword stored in input_vectector
Step 4: Perform the dot (.) operation between each category files and simil matrix.
Step 5: All the coordinates of the same files store in an individual_vector matrix.
Step 6: Calculate the distance between input_vector and each indivisual_vector.
Step 7: Identify the minimum distance and fetch keyword from the minimum distance location from each categorical_data_files.
Step 8: Store all words in text_list matrix.
The text_list matrix gives the all combination of the input keyword.

4. Searching

In this module load the final_cluster file which contains the final clusters. Then search all the possible keywords which are stored in text_list matrix. To do the above we have to follow some steps which are described below:
Step 1: Load the final_cluster files.
Step 2: Search each word in text_list matrix in each of the categorical_data_files and output store in a finput_vector matrix.
Step 3: Calculate the distance between centroid_cluster matrix and finput_vector matrix.
Step 4: Find out the minimum distance and location of it.
Step 5: Fetch files from this location of final_cluster and display them.

5. Result Analysis

The objective of our searching tool is to provide the correct result all the time even if the input keyword is wrongly spelled. Many existing searching tools are able to retrieve the result when user enters the keyword instead of the name of the file which user wants to search but not give the satisfactory result for spelling mistake.

In this paper five coordinates system was used. Instead of five if six coordinates system can be used better result can be obtained. The file names are the decimal number starting from 1 to 925.

Time Comparison:

Search keyword	Time Required (Five coordinate system)	Time Required (Six coordinate system)	Displayed File Names	
			1st File Name	2nd File Name
(i)INDIA(correct spelling)	.141000 secs.	.250000secs.	23	50
(ii)INDEA(incorrect spelling)	.125000 secs.	.21800secs.	23	No files
(i)JAPAN(correct spelling)	.106644 Secs.	.106644 Secs.	453	762
(ii)JEPAN(incorrect spelling)	.11097865Secs.	.11097865Secs.	453	No files

Table 2: Experimental Results

5. Conclusion

In this paper we try to design the search tool which is based on "Meta-Clustering Technique" which is capable for searching a file using misspelled keyword from the static database. This method can be extended on dynamic database.

RERERENCES

1. D.Buettner and B.Markscheffel, *Ein Vergleich ausgewählter Desktop-Suchmaschinen* Ilmenauer Beiträge zur Wirtschafts informatik Nr. 2011-02, Ilmenau, 2011.
2. E. Folmer, *Software Architecture – Analysis of Usability*, Enschede, 2005.
3. S.Mandal and G. Silakari, Meta-Clustering Technique for Document Searching *TRACE-2010*, Feb.24th 2010.
4. J.Han and M.Kamber, *Data Mining: Concepts and Techniques*, Morgan Kaufman, 2001.
5. Y.Zhao and G.Karypis, *Criterion Functions for Document Clustering: Experiments and Analysis*, UMN CS01-040, 2001.
6. C.-H.Cheng, W.-C.Fu and Y.Zhang, Entropy- Based Subspace Clustering for Mining Numerical Data, In *Proc. ACM SIGKDD*, 1999.
7. C.C.Aggarwal, C.M.Procopiue, J.L. Wolf, P.S.Yu and J.S.Park, Fast algorithm for Projected Clustering, in *Proc. SIGMOD l Conference*, Philadelphia, PA 1999.
8. G.Karypis, E.H.Han and V.Kumar, CHAMELEON: A hierarchical clustering algorithm using dynamic modeling, *Computer*, 32(8) (1999) 68-75.
9. A.K.Jain, M.N.Murty and P.J.Flyn, Data Clustering: A Review, *ACM Computing Surveys*, 31(3) (1999).