ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2015

On-Lib: An Application and Analysis of Fuzzy-Fast Query Searching and Clustering on Library Database

Ashritha K.P, Sudheer Shetty

4th Sem M.Tech, Dept. of CS&E, Sahyadri College of Engineering and Management, Adyar, Mangalore, India

HOD, Dept. of CS&E, Sahyadri College of Engineering and Management, Adyar, Mangalore, India

ABSTRACT: Instant searching is an emerging technique which is used by most of the search engines. It will return the results for a user query instantly, i.e., before the user types in the entire query. This technique will improve the user experience by predicting the results for the query by matching the keywords in the query. Here the main constraint is the time to process each keyword and display the result instantly. This paper uses two techniques to meet the time requirement and predict most relevant answers. Instead of providing mixture of results for a query, clustering is done on the query results based on the similarity measure. Clustering at sentence level will group the sentences based on the similarity measure. Clustering technique used here will improve the overall performance of searching algorithm.

KEYWORDS: Clustering, Fuzzy Logic, Instant searching, Sub-String matching

I. INTRODUCTION

Problem Statement: In this paper, the following problem is analysed: how to include clustering and relationship between the keywords and sentences into a fast-fuzzy searching algorithm, so that most relevant search results are provided to the user. The database considered is a library database containing names of books along with author name and book IDs.

In instant search technique, when the user types a query partially, the answers are returned immediately. For example: In an online library interface, if the user types: "op", it should return words such as "operate", "operations", "operating" etc. More often the users expect that the suggestions for their partial query must appear in no time, before the entire query has been typed in. This will help the users to find relevant answers instantaneously with less effort [1]. But the users are often likely to make typing mistakes in search queries. The reasons for such mistakes can be: touch screens or small keyboards on mobiles/tablets, lack of care, or insufficient knowledge about the data. Providing the results which will match exactly with keyword typed is difficult or inefficient in this case. Fuzzy searching can be a solution to this. It will return the answers that will partially match with the keyword typed by the users. The computational challenge in this case is the high-speed requirement.

Wherever the user types a query, he should not experience any delay. He should get very quick response. In order to get a quick response, it is necessary to answer each query within milliseconds [2]. The more complex challenge to the server is to provide high quality answers that will match with the information the user actually needs. Search queries typically contain correlated keywords, and answers that have these keywords together are more likely what the user is looking for. For example, if the search query is Sachin Tendulkar, the user is most likely looking for the records containing information about the cricket player Sachin Tendulkar, while documents containing Sachin Raikar would be less relevant.

Sentence clustering has variety of applications such as classification of documents, organization of documents, checking the contradictions among set of documents etc. In hard clustering techniques such as k-means, k-medoids etc., a pattern belongs to a single cluster. But fuzzy clustering algorithms allow a pattern to belong to more than one cluster at the same time. A novel fuzzy clustering algorithm that operates on relational input data; i.e., data is represented as a

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2015

square matrix of pairwise similarities between data objects, is used in this paper. This algorithm can be used in variety of text mining tasks.

In this paper, a fuzzy based sentence clustering and instant searching are studied where clustering would organise the records in a well-mannered fashion based on sentence similarity measure. The main performance requirement in case of instant searching is time and space efficiencies. String matching algorithms are used to handle time efficiency and space issues are handled by providing top k-query relevant answers.

II. RELATED WORK

In Auto-prediction, all the possible queries that user might type next are provided by matching the keyword user has typed in. But most of the systems take a phrase (user query) with multiple keywords as single string. So even if there is a related suggestion, but keywords are not consecutive, then prediction may not be accurate. This can be a drawback for such systems.

There are two basic approaches for fuzzy search: gram-based approach and a trie based approach. In gram based approach, sub-string matching techniques are used in order to find word to word similarity [3][4][5], where as in trie based approach, indexes are used to find the matching words [6][7].

Fuzzy search will provide all the records that match with the partial/complete query user has typed in. But as the size of the database increases, number of matching records may also increase. Therefor the complexity also increases. The solution to this is providing the top answers that match with the user query.

This paper uses clustering technique to choose the top relevant answers that match with the user query. Instead of displaying mixture of query results, the results are chosen from the cluster which has maximum similarity with the user query.

There are various fuzzy relational clustering techniques out of which the first one was Relational Fuzzy c-means clustering algorithm [8]. It uses the Euclidean distance as a relation measure between the data points. Non-Euclidean relations have to be transferred to Euclidean by using a constant β. But during this transformation there will be loss of certain information. Methods based on k-medoids [9] are also available. But they are sensitive to initial cluster centres. Therefore they have to be executed multiple times with multiple initializations. So they are time consuming.

The clustering technique used in this paper is Fuzzy based clustering on a relational database. It uses similarity between two records in order to compute cluster membership values. The membership value for each record represents the degree by which the record belongs to a cluster.

III. PROPOSED ALGORITHM

*Preliminaries:*The data set considered is set of records, $R=\{r_1, r_2, \ldots r_n\}$ where n is the total number of records in the dataset. The records are clustered and membership of each record in each cluster is computed. The user query $Q=\{q_1,q_2,\ldots q_n\}$ is the query given by the user where q_i is the keyword. The answers to the query are the records that satisfy following conditions: 1.The record which has matching keyword, 2.The record with matching substring as of the keyword.

The similarity between two words can be measured in various ways such a cosine similarity, Jaccard similarity or Levenshtein distance. Here we use Levenshtein distance as a similarity measure. It gives the minimum number of insertions, deletions or substitutions needed to convert one string to other.

Description of the Proposed Algorithm:

Clustering:

In the first step, the clustering algorithm is executed by taking the records in the database as the input. The number of clusters C is also given as input. The algorithm returns the membership value of each record in each cluster. This determines amount by which a record belongs to a cluster. In general, the similarity between the objects within a cluster

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2015

is maximum and Similarity between objects belonging to two different clusters is minimum. In other words intra cluster similarity is maximum and inter cluster similarity is minimum.

The similarity between the objects is stored in the similarity matrix $S_{i,j} = \{ s_{i,j}\}$ where $s_{i,j}$is the similarity between sentence i and j. Here the similarity is measured in terms of number of words common in sentences i and j. Other similarity measures can also be used. During the initialization step, cluster membership values $p_{i,j}$ are assigned randomly and are normalised. The weights are computed as[10]:

$$w_{i,j}^m = s_{i,j} \times p_i^m \times p_j^m \quad (1)$$

Where,

$w_{i,j}^m$ = weight between objects i and j in cluster m.

$s_{i,j}$ = similarity between object i and j.

p_i^m andp_j^m are the membership values of objects i and j in cluster m respectively.

The page rank value is computed as[10]:

$$PR_i^m = (1\text{-}d) + d \times \sum_{j=1}^{N} \left(w_{i,j} \frac{PR_j^m}{\sum_{k=1}^{N} w_{j,k}^m} \right) \quad (2)$$

Where, d is the damping factor which is set to 0.8 or 0.9[11].

The membership values are updated using the equation given below.[10]

$$p_i^m = \frac{(\pi_m \times PR_i^m)}{\sum_{j=1}^{C} (\pi_j \times PR_i^j)} \quad (3)$$

Where, π_m is the mixing coefficient. It is computed as [10]:

$$\pi_m = \frac{1}{N} \sum_{i=1}^{N} p_i^m \quad (4)$$

Searching:

The first step in searching is to identify all the valid phrases in the query types by the user. The valid phrase is the one which matches with the phrases in the database. To match a valid phrase, we need to do all possible combinations of keywords in the query. Yet all the combinations may not be valid. Fortunately, the number of keywords in a query is not large. i.e., web search usually includes 2-4 keywords [12]. To generate valid phrases string matching technique is used.Once all the valid phrases have been matched, segmentations have to be generated. Segmentation includes breaking a valid phrase into different segments.

Performance measures:

Performance of Clustering technique can be computed using unsupervised techniques or supervised techniques. In case of unsupervised technique, a measure used is Partition Entropy Coefficient [13]. It is defined as:

$$PE = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{|L|} u_{ij} \log_a u_{ij} \quad (5)$$

Where N is the total number of records, L is the number of clusters formed and u_{ij} is the membership value of sentence i in cluster j. Value of PE decides the crispness of clustering. As the value increases, clustering is crisper.

IV. PSEUDO CODE

The pseudo code for clustering algorithm used is given below:

Procedure Initialization (i,m)

--i is the number of records

--m is the number of clusters

Step 1: Choose p[i][m] as a random number between 0 and 1

Step 2: Normalize p[i][m].

Step 3: Provide equal priority to all the clusters.

Procedure Expectation_Maximization (p,i,m)

--p is the membership matrix

Step 4: Compute weight for each sentence w[i][m] using (1)

Step 5: Compute page rank PR[i][m] using (2)

Step 6: Calculate new membership values using (3)

Step 7: Update mixing co-efficient using (4)

This algorithm gives the output as membership values which indicate the membership of sentence i in cluster m. Two types of clustering are possible. First one is crisp or hard clustering which assigns an object completely to one

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2015

cluster at a time. Second one is fuzzy clustering which assigns a sentence to each cluster partially. Membership values plays important role in case of fuzzy clustering technique as it computes the percentage by which each sentence belongs to a cluster. Once clustering is completed we proceed for searching step.

The pseudo code for doing the segmentations is given below:

Procedure Segmentation (V_i)

--V_i is set of valid phrases

Step 1: Initialise Segmentation vector to null.

Step 2: For each valid phrase, generate all valid segmentations

Step 3: Append each segmentation to segmentation vector.

Each of the segments is then compared with the keywords in the dataset. The record which has minimum Levenshtein(edit) distance is given a higher rank. This step is repeated till we get top-k query results. The user can then choose the required data from set of results. All the related data can now be taken by selecting the cluster to which the record selected by the user belongs.

V. RESULTS

The clustering algorithm was run using extracts from famous news articles dataset dataset in order to measure the performance and the PE value was found as 1.40[10]. To demonstrate the more general use of algorithm, we used on a sample library dataset. The data set will contain set of Text books. Using a fuzzy based clustering algorithm on a relational database, all the Records will be grouped based ona similarity measure and clusters with similar records will be generated. Further, since a fuzzy basedapproach is used to form clusters, clusters can be overlapped.In the front end, user will interact with the system using a well-defined interface which provides an option for entering the queries. The user can search a book from library database where he will get instant suggestions based on the keywords typed. When the user types in each character, suggestions will be generated. This helps the user to select the appropriate record. This is done usinga Fuzzy based Instant search method. Further, even though the user search keywords does not match completely with any sentence inthe set of documents, instant search algorithm is expected to return the results of most closely relatedkeywords in the database. This ensures that even if the end-user has very little knowledge regarding thedatabase, he can still search his topic of interest. Figure 1 Shows an Example for instant fuzzy search onIrvine People Dictionary [14] which shows that even though if the user types in different characters, mostappropriate answers in the database are displayed. This can be termed as natural language processing.

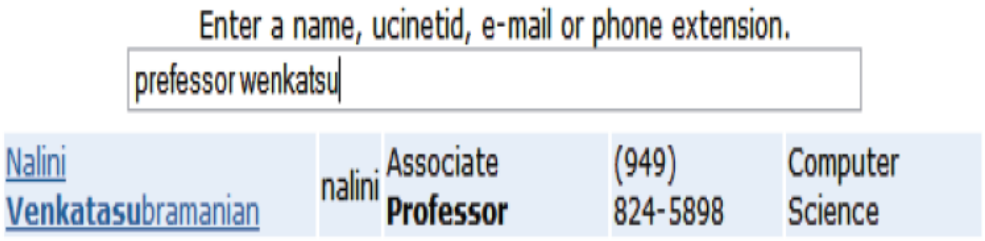

Fig 1: An example for instant fuzzy search on UC Irvine People Dictionary[14]

The fast-fuzzy searching algorithm is implemented on the Library Database where the user can type in the title of the book he is looking for. The database contains 1000 records. Using the fast fuzzy technique, the user is expected to get high quality result within a short period of time.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2015

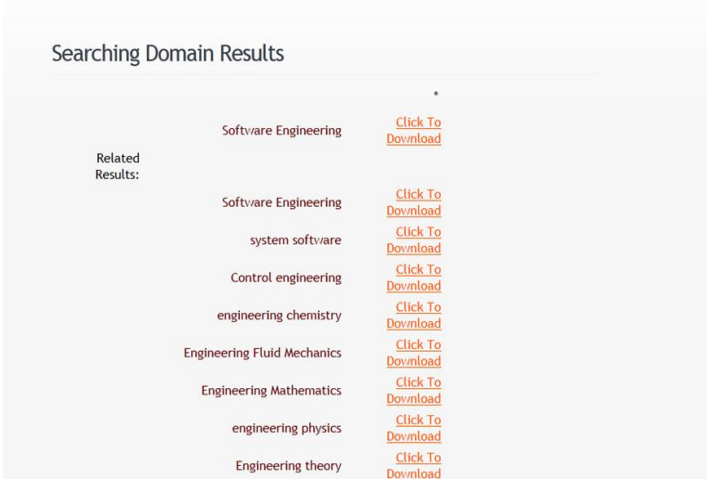

Fig 2: Sample Result for Fast-Fuzzy Search algorithm.

The Result shown above is the search result for a book name typed by the user. The admin will upload the books to the library database. He then applies the clustering algorithm on the database by taking the titles of the book as an input. When the user searches a book of his interest, he will get the results if the book is found in the database. He will also get the related results based on the clustering results.

VI. CONCLUSION

A Fuzzy based Clustering algorithm on relational data can be used for sentence level clustering. This will help in grouping the similar sentences into single cluster. Since a fuzzy based approach is user overlapping clusters can be determined. Efficient fast-fuzzy search algorithm based on substring matching is used to compute relevant answers based on proximity information ranking.The important computational requirement of instant searching is space and time efficiencies. Both these requirements are expected to be met by using the above mentioned techniques.

ACKNOWLEDGEMENT

We express sincere gratitude to Management and Staff of Sahyadri College of Engineering and management, Mangalore for proving a platform to conduct research and experimentation.

REFERENCES

1. Cetindil, I., Esmaelnezhad, J., Li, C., and Newman, D.,"Analysis of Instant Search Query Logs", WebDB, pp. 7-12, 2012.
2. Miller, R. B., "Response time in man-computer conversational transactions", Proceedings of the December 9-11, fall joint computer conference, part I ,pp. 267-277, 1968.
3. Bast, H., and Weber, I., "Type less, find more: fast autocompletion search with a succinct index", Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 364-371, 2006.
4. Bast, H., Chitea, A., Suchanek, F., and Weber, I., "Ester: efficient search on text, entities, and relations", Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval,pp. 671-678, 2007.
5. Bast, H., and Weber, I., "The CompleteSearch engine: Interactive, efficient, and towards IR & DB integration", Untitled Event-University of Wisconsin/Computer Science Department, pp. 88-95, 2007.
6. Ji, S., Li, G., Li, C., and Feng, J., "Efficient interactive fuzzy keyword search", Proceedings of the 18th international conference on World wide web, pp. 371-380, 2009.
7. Chaudhuri, S., and Kaushik, R., "Extending autocompletion to tolerate errors", Proceedings of the ACM SIGMOD International Conference on Management of data, pp. 707-718, 2009.
8. Hathaway, R. J., Davenport, J. W., and Bezdek, J. C., "Relational duals of the c-means clustering algorithms", Pattern recognition, Vol.22, Issue 2, pp. 205-212,1989.
9. Hastie,T., Tibshirani,R., and Friedman,J., "The Elements of Statistical Learning: Data Mining, Inference, and Prediction", Springer, 2001.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2015

10. Skabar, A., and Abdalgader, K., "Clustering sentence-level text using a novel fuzzy relational clustering algorithm", Knowledge and Data Engineering, IEEE Transactions on, Vol.25, Issue 1, 62-75,2013.
11. Kaufmann, L., and Rousseeuw, P. J., "Finding groups in data", New York: J. Wiley & Sons, 1990.
12. Arampatzis,A., and Kamps,J., "A study of query length",*SIGIR*, pp. 811–812,2008.
13. Bezdek, J. C., "Mathematical models for systematics and taxonomy", Proceedings of eighth international conference on numerical taxonomy ,Vol. 3, pp. 143-166,1975.
14. Cetindil, I., Esmaelnezhad, J., Kim, T., and Li, C., "Efficient instant-fuzzy search with proximity ranking", In Data Engineering (ICDE), IEEE 30th International Conference,pp. 328-339,2014.