Optimal keyword search using Rocchio Algorithm over cloud data

Ms. Nikita M. Talhar
SGBAU, Amravati
Maharashtra, India.
Email : niki.talhar@gmail.com

Dr. V.M. Thakare
SGBAU, Amravati
Maharashtra, India.
Email : vilthakare@yahoo.co.in

ABSTRACT

Cloud has large storage capacity and flexible accessibility. By outsourcing the sensitive data to a cloud server, individuals and enterprises are relieved from the burden of local data management and maintenance. But retrieving the relevant documents and searching semantic query over large database is still challenging. This paper proposes a novel "Clustered Rocchio Search using SemanticLib" framework. The proposed framework will retrieve the relevant document from database the user is searching for. The idea behind this approach is to search semantic query keywords with highest ranking. This paper presents the concept of clustering for efficiently access the relevant information according to domain. To search the information related to keyword query Rocchio Algorithm is used. The result contains required information as semantics-based searching is done with the help of Semantic Library. The proposed framework is simple, efficient and reduces searching time.

***Index Terms* — Clustering, Rocchio Algorithm, Semantic Library, Keyword search.**

I) INTRODUCTION

Cloud computing is a subversive technology that is changing the way IT hardware and software are designed and purchased. As a new model of computing, cloud computing provides abundant benefits including easy access, decreased costs, quick deployment and flexible resource management, etc. Enterprises of all sizes can leverage the cloud to increase innovation and collaboration. Cloud computing has emerged as a new enterprise IT architecture [1]. With the growing popularity of cloud computing, huge amount of documents are outsourced to the cloud for reduced management cost and ease of access. Keyword search is a proven and widely accepted mechanism for querying in textual document systems and the World Wide Web [2], [3]. The database research community has recently recognized the benefits of keyword search and has been introducing keyword-search capabilities into databases. Most information systems today rely on a large number of data sources [4]. It needs to combine similar data for efficient search. Information can be structured or unstructured. Queries on structured data are issued assuming that correct specification of the user information need exists and that answers are perfect. To proposed new method for keyword search checking the performance evaluation of existing techniques is very important. Unless knowing the drawback of previously proposed techniques or approaches it is difficult to propose new techniques [5]. So performance evaluation is also important factors like throughput, memory utilization, execution time has relatively little impact on system.

This paper proposed a novel framework i.e. **"Clustered Rocchio Search using SemanticLib".** This paper presents the concept of clustering for efficiently access the relevant information according to domain. Rocchio Algorithm is used to search the information related to keyword query with highest ranking. This algorithm is simple and has ability to incorporate term weights. The result contains required information as semantics-based searching is done with the help of Semantic Library. The proposed framework is simple, effective and the accuracy of result is improved.

This paper is organized as follows. **Section I** contains Introduction of this paper. In **Section II** discussed Background. **Section III** introduced previous work done. **Section IV** explains existing methodologies. In **Section V** discussed existing framework and analysed it. **Section VI** presents the overview of the Clustered Rocchio Search using SemanticLib framework. Its outcome possible results are analysed in **Section VII**. **Section VIII** concludes this paper. Finally **Section IX** presents future scope.

II) BACKGROUND

Many studies on keyword search have been done to develop the efficient route and speedily retrieve information in recent past years. Keyword Search for Service-based Systems (KS3) [1] approach is proposed for building Service-based Systems (SBSs) based on keyword search. It automates and integrates system planning, service discovery and service selection. Proposed approach allows system engineer without details knowledge of SOA for searching with quality constraint and optimization goals. This approach is proposed to save efforts. A novel method known as ProMiSH (Projection and Multi Scale Hashing) [2] is proposed for fast processing of Nearest Keyword Set (NKS) queries. ProMiSH-E and ProMiSH-A is used to retrieve optimal top-k results and provide time and space efficiency respectively. The proposed method uses random projection and hash-based index structures. It achieves speedup and high scalability. Keyword Nearest Neighbor Expansion (keyword-NNE) [3] is proposed to reduce the number of candidate keywords generated. It investigated generic version of mCK query known as Best Keyword Cover (BKC). It considers inter-objects distance and keyword ratings. It applies different processing strategies such as searching local best solution for each object in query keyword. It generates lesser number of new candidate keyword cover. The proposed method solved the problem of keyword query routing in keyword search over large number of linked and structured data sources. It represents relationship between keywords and data elements. First they are constructed for entire collection of linked source and after that grouped as elements of compact summary known as set-level keyword-element relationship graph (KERG) [4]. It is necessary to address the scalability. The keyword search performance is improved without affecting result quality. The performance evaluation [5] is done on various keyword search techniques. It checks whether it provides efficient performance for realistic retrieval task. The analysis done in this paper is observed that these factors have little impact on performance. It also checks for memory consumption.

III) PREVIOUS WORK DONE

Qiang He et al (2017) proposed Keyword Search for Service-based Systems which helps system engineers for searching web services for building SBSs without having detailed knowledge of SOA. KS3 works with directed graph data. Nodes are used for web services and edges represent service composability. Constraint Optimization Problem model answer queries for building SBSs. It supports normal, constraint and optimal queries [1].

Vishwakarma Singh et al (2016) proposed a method ProMiSH which uses random projection and hash-based index structures to provide solution to the problem of top-k NKS search in multidimensional dataset. It is faster than state-of-the-art tree-based techniques. ProMiSH-E uses inverted indexes and hashtables to perform localized search. ProMiSH-E explores subsets using a pruning-based algorithm. ProMiSH-A provides better space and time efficiency [2].

Ke Deng et al (2015) observed the increasing importance and availability of keyword rating for better decision making in object evaluation. Scalable keyword-NNE [3] algorithm is proposed which selects one query keyword as principal query keyword. The local best solution i.e. local best keyword cover is computed for each principal object. The in-depth analysis shows that number of candidate keyword cover is reduced in keyword-NNE.

Thanh Tran et al (2014) proposed method to compute top-k routing plans according to potential. IR-ranking incorporates relevance at the level of keywords. It introduced multilevel relevance model in which elements are considered as keywords, set of entities and relationships between elements at same level and at different level. Scoring mechanism computes relevance of routing plans on the basis of scores [4].

Joel Coffman et al (2014) conducted an independent evaluation of runtime performance of different search techniques proposed previously. Some of the existing search techniques does not perform well on large databases and needed inordinate amount of memory. This paper investigated that many parameters are loosely correlated and have lack of meaningful relationship [5].

IV) EXISTING METHODOLOGIES

A. Keyword Search for Service-based Systems (KS3):

KS3 is independent of the approach used for generation of data graph and runs on any data graph. KS3 constructs an inverted index for a data graph G. where for every keyword in G the nodes covering the keyword are stored in this index. It first locates nodes that contain individual keywords and then finds the set of nodes using inverted index. For answering constraint queries it must be exact group strainer tree that contains all keywords and check quality constraints. Optimal query is a constraint query with optimization goals. It checks for optimal system reliability, optimal system throughput, optimal system cost and optimal system utility. KS3 uses unidirectional breadth-first algorithm to travel graph and allow using pre-specified system structures.

B. ProMiSH (Projection and Multi Scale Hashing):

Index structure of ProMiSH-E consists of two main components i.e. Inverted Index and Hashtable-Inverted Index Pairs (HI). Search algorithm for ProMiSH-E is presented to find top-k results for NKS queries. The index structure and flow of execution of ProMiSH is shown below.

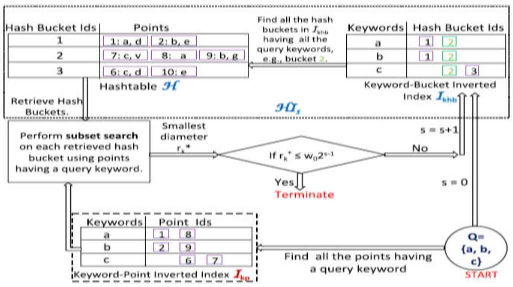
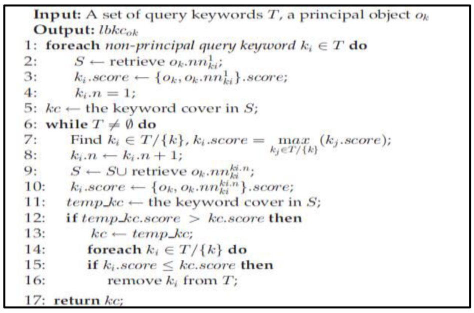

Hash Bucket Ids
Points
1
2
3
1: a, d
2: b, e
7: c, v
8: a
9: b, g
6: c, d
10: e
Hashtable $\mathcal{H}$
Find all the hash buckets in I_{khb} having all the query keywords, e.g., bucket 2.
Keywords
Hash Bucket Ids
a
b
c
1
2
3
Keyword-Bucket Inverted Index I_{khb}
Retrieve Hash Buckets.
$\mathcal{HI}_s$
Perform subset search on each retrieved hash bucket using points having a query keyword.
Smallest diameter r_k^*
If $r_k^* \le w_0 2^{s-1}$
No
s = s+1
Yes
Terminate
s = 0
Keywords
Point Ids
a
b
c
1
8
2
9
6
7
Keyword-Point Inverted Index I_{kp}
Find all the points having a query keyword
Q= {a, b, c}
START

Fig1. Index structure and execution flow of ProMiSH

ProMiSH-A provides near-optimal results. It differs from ProMiSH-E because it partitions projection space into non-overlapping bins of equal width, where ProMiSH-E partitions projection space into overlapping bins. Hence it is time and space efficient.

C. Keyword Nearest Neighbour Expansion (keyword-NNE):

The proposed approach uses a three-dimensional R*-tree called keyword rating R*-tree (KRR*-tree) is used. The ranges of spatial and keyword rating dimensions are normalized into [0, 1]. It uses principal query keywords. The objects associated with it are known as principal objects. For each principal object computing local best keyword cover (LBKC) is important. LBKC algorithm is explained below.



```
Input: A set of query keywords $T$, a principal object $o_k$
Output: $lbkc_{ok}$
foreach non-principal query keyword $k_i \in T$ do
    $S \leftarrow$ retrieve $o_k.nn^1_{ki}$;
    $k_i.score \leftarrow \{o_k, o_k.nn^1_{ki}\}.score$;
    $k_i.n = 1$;
$kc \leftarrow$ the keyword cover in $S$;
while $T \neq \emptyset$ do
    Find $k_i \in T/\{k\}$, $k_i.score = \max_{k_j \in T/\{k\}}(k_j.score)$;
    $k_i.n \leftarrow k_i.n + 1$;
    $S \leftarrow S \cup$ retrieve $o_k.nn^{ki.n}_{ki}$;
    $k_i.score \leftarrow \{o_k, o_k.nn^{ki.n}_{ki}\}.score$;
    $temp\_kc \leftarrow$ the keyword cover in $S$;
    if $temp\_kc.score > kc.score$ then
        $kc \leftarrow temp\_kc$;
        foreach $k_i \in T/\{k\}$ do
        if $k_i.score \le kc.score$ then
            remove $k_i$ from $T$;
return $kc$;
```

Fig2. Local Best Keyword Cover Algorithm

KRR*-tree is browsed using depth-first strategy. In keyword-NNE algorithm principal objects are processed in blocks and they are indexed using KRR*-tree. The keyword cover with highest score is maintained in memory. Here the best-first strategy is applied and large memory requirements are avoided.

D. Keyword-Element Relationship Graph (KERG):

To solve the problem of keyword query routing it uses graph based data model which characterized individual data sources. The problem of keyword query routing is to find top-k query routing plans. It works on element-level data graph and set-level data graph. Element-level entities are associated with set-level elements. It constructs element-level keyword-element relationship graph [4]. It can retrieve connections and check whether the corresponding entities are connected and at the end extracts source information for construction of keyword RP. The rank of keyword ranking plan RP is computed as an aggregation which returns the k-best ranked routing graphs in RP. The proposed summary model is used to group keywords and element relationship at the level of sets and multilevel ranking scheme is developed to incorporate relevance at different dimensions.

E. Performance Evaluation:

To proposed new method for keyword search checking the performance evaluation of existing techniques is very important. Unless knowing the drawback of previously proposed techniques or approaches it is difficult to propose new techniques. This paper observed the performance evaluation of several existing keyword search techniques. It uses two matrices for measurement of runtime performance. First one is execution time and second is response time. Execution time is a time elapsed from issuing a query until an algorithm terminates. Response time is a time elapsed from issuing the query until some results have been returned. It depends on number of search terms, collection frequency, results size and retrieval depth [5].

V) ANALYSIS AND DISCUSSION

Keyword Search for Service-based Systems approach helps system engineers for searching web services for building SBSs without having detailed knowledge of SOA. KS3 can handle multiple keywords in one query. System engineers do not have to enter the keywords in a specific order. It saves time and improve throughput [1].

ProMiSH [2] is proposed for the solution to the problem to find nearest keyword set search in multi-dimensional dataset. It achieves high scalability and speedup. It can be useful in multi-dimensional dataset. It can work with less memory and less indexing time. It is time and space efficient. But Weight of the keyword is not assigned.

In proposed algorithm i.e. keyword-NNE [3], the number of candidate keyword covers generated is significantly reduced so that performance does not decrease as the number of query keywords increase. It is suitable in inter-objects distance as well as the keyword rating of objects. But the drawback is that Keyword-NNE algorithm implementation is more complex than baseline algorithm.

The proposed model [4] solved the problem of keyword query routing. It improves the performance of keyword search, without compromising its result quality. Keyword query routing can be employed when the subject of interest is not necessarily results but sources that match some information needs.

The performance evaluation [5] is done on various keyword search techniques. It evaluated relational keyword search techniques and evaluated them with regard to their search effectiveness. It uses realistic data sets and realistic queries.

Keyword Search Approaches	Advantages	Disadvantages
Keyword Search for Service-based Systems (KS3)	It helps system engineers to search web services for building SBSs without having detailed knowledge.	KS3 constraint method and optimal method are not always able to find a solution due to the quality constraints.
ProMiSH	It find nearest keyword set search in multi-dimensional dataset.	Weight of the keyword is not assigned.
Keyword Nearest Neighbour Expansion	The number of candidate keyword covers generated is significantly reduced.	Algorithm implementation is more complex.
Keyword-Element Relationship Graph (KERG)	It reduces the high cost of processing keyword search queries over all sources. It produces routing plans.	Keyword query routing can be employed when the subject of interest is not necessarily results.
Performance evaluation	It evaluated relational keyword search techniques with regard to their search effectiveness.	Only performance evaluation is done on realistic query workload.

Table 1: Comparison between different keyword search approaches

VI) PROPOSED METHODOLOGY

Clustered Rocchio Search using SemanticLib

Cloud computing has become a promising technology due to its impressive features, i.e., large storage capacity and flexible accessibility. By outsourcing the sensitive data to a cloud server, individuals and enterprises are relieved from the burden of local data management and maintenance. But retrieving the relevant documents is also challenging. Sometimes user is unaware about how to precisely express their queries to access relevant information. There is also a problem of searching semantic query over large database. So to solve this problem this paper proposed a novel **"Clustered Rocchio Search using SemanticLib"** framework.

The proposed framework will retrieve the relevant document from database the user is searching for. It is very difficult and time consuming to search over large database, so here it uses the concept of clustering for efficient access to relevant information. Clustering is a process of partitioning a set of data into a set of meaning sub-classes known as clusters. Clustering can be domain related. It reduces searching time. To search the information related to keyword query Rocchio Algorithm is used. The Rocchio Algorithm is used in information retrieval system. This algorithm results according to highest ranking of document. This algorithm is simple and has ability to incorporate term weights. It does not require any other model that has to justify the use of a weight. It can measure similarities between document and queries, documents and documents. But if there is a variation in word or need to search synonym keyword then Rocchio algorithm vector space model will treat them differently. So to overcome this limitation this paper proposed new approach that is Semantic Library. Semantic Library is a library of keywords which stores the keywords with its variation words or synonyms keywords. So that searching will be effective over database. This proposed framework provides effective way to search relevant information over database, it saves searching time since it uses the clustering and it is simple.

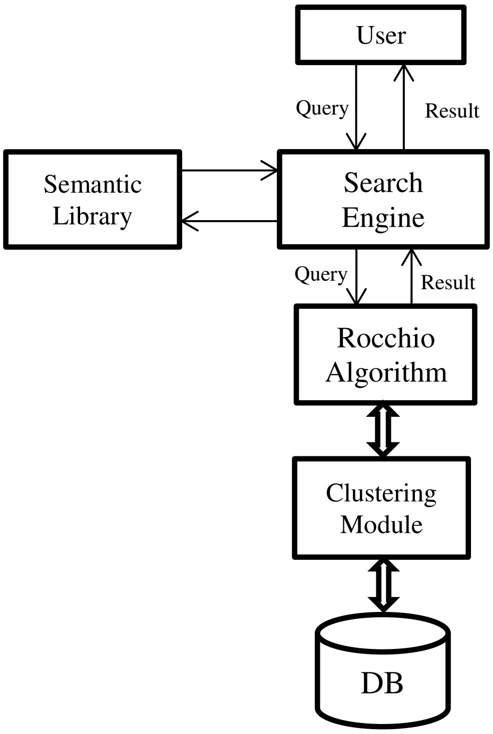


Fig3. Clustered Rocchio Search using SemanticLib Framework

The above diagram describes the working of proposed framework. User first input query. Query goes to search engine for further processing. Search Engine is connected to Semantic Library. Semantic Library is a library of keywords which stores keywords with its variation words or synonyms keywords. If any keyword found in Semantic Library which is also present in user query then update the query and add the new keyword in user query. If nothing is found then don't change query and return. This new query will be given to the Rocchio Algorithm. Rocchio algorithm searches in database for relevant documents. In clustering module, clustering is done on database based on domain. Document ranking is done in Rocchio algorithm. Generate the response for the current information retrieval process.

Clustered Rocchio Search using SemanticLib Algorithm:

```
1. Take input from the user
2. Check for the variation of words or synonyms
   keywords in Semantic Library
       if found
           Update the query
       else
           Don't update the query
```

```
3. Call Rocchio Algorithm
4. End
```

Rocchio Algorithm:

```
1. Take input {i1…….in}
2. Start the query processing and clustering
3. Shaping the modified vector
4. Creating associated weights (a, b, c)
5. Values for b and c should be incremented or
   decremented proportionally to the set of
   documents classified
6. Information retrieval process
   The list of documents {d1….dn}
   while(d1==i1)
   {
           Result= Dr
   }
   Else
   {
           Result=Dnr
   }
7. Assigning ranking to the documents
   {d1………dn} according to relevance
8. Generate the response for the current
   information retrieval process
```

Where,

i1…in = input query

d1…..dn = processed documents

a, b, c = associates weight of document

Dr = sets of vectors containing the coordinates of related documents

Dnr = sets of vectors containing the coordinates of non-related documents

VII) OUTCOME AND POSSIBLE RESULT

This paper proposed a “Clustered Rocchio Search using SemanticLib” framework providing query results that are relevant to the user information needs. Since proposed framework uses the concept of clustering the searching time is reduced. The performance analysis of proposed framework is depends upon how efficiently it will search domain specific query keywords. The Rocchio algorithm assigns ranking to the documents according to relevance. The result contains required synonyms information as semantics-based searching is done. The result demonstrates that proposed framework provides optimal way for keywords search over cloud data.

VIII) CONCLUSION

This paper describes various query keyword searching frameworks such as Keyword Search for Service-based Systems, ProMiSH, Keyword Nearest Neighbor Expansion, keyword-element relationship graph. This paper proposed “Clustered Rocchio Search using SemanticLib” framework which is used to search relevant documents for user query efficiently. The concept of Semantic Library is proposed which stores the keywords with its variation words or synonyms keywords. Since it uses the concept of clustering the searching time is reduced. Rocchio algorithm results according to highest ranking of document. The proposed framework is simple, efficient and reduces searching time.

IX) FUTURE SCOPE

Further analysis and efforts required to search keyword queries over random dataset. Query workload can be reduced to provide greater consistency of results.

REFERENCES

[1] Qiang He, Rui Zhou, Xuyun Zhang, Yanchun Wang, Dayong Ye, Feifei Chen, John C. Grundy, and Yun Yang, “Keyword Search for Building Service-Based Systems,” IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, vol. 43, no. 7, pp. 658-674, JULY 2017.

[2] Vishwakarma Singh, Bo Zong, and Ambuj K. Singh, “Nearest Keyword Set Search in Multi-Dimensional Datasets,” IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, vol. 28, no. 3, pp. 741-755, MARCH 2016.

[3] Ke Deng, Xin Li, Jiaheng Lu, and Xiaofang Zhou, “Best Keyword Cover Search,” IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, vol. 27, no. 1, pp. 61-73, JANUARY 2015.

[4] Thanh Tran and Lei Zhang, “Keyword Query Routing,” IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, vol. 26, no. 2, pp. 363-375, FEBRUARY 2014.

[5] Joel Coffman, and Alfred C. Weaver, “An Empirical Performance Evaluation of Relational Keyword Search Techniques,” IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, vol. 26, no. 1, pp. 30-42, JANUARY 2014.