IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Volume-4, Issue-4 E-ISSN: 2347-2693

A Novel Framework For Enhancing Keyword Query Search Over Database

Priya Pujari[1*] and Arti Waghmare[2]

[1, 2] *Department of Computer Engineering,*

Dr.D.Y.Patil School of Engineering and Technology, Savitribai Phule Pune University, Pune, India

Available online at: www.ijcseonline.org

Received: Mar/23/2016 Revised: Apr /01/2016 Accepted: Apr/16/2016 Published: Apr/30/2016

***Abstract*—** Data that exists in fixed field in a record is called as structured data and putting away such data into database is broadly expanding to strengthen keyword query yet result lists do not give successful responses to keyword query and subsequently it is hard from user's point of view. It is useful to grasp such kind of queries which gives results with low positioning. Here we determine identification of such queries to discover power of search performed in reply of query and characteristics of such hard query is identified by considering building blocks of the database and result list. One applicable issue of database is the existence of missing data and it can be resolved by imputation. Here an inTeractive Retrieving-Inferring data imPutation method (TRIP) is utilized which accomplishes retrieving and inferring in successive manner to fill the missing attribute values in the database. TRIP can also analyze optimal scheduling scheme in Deterministic Data Imputation (DDI). Filling missing values in such successive manner, we can improve the precision of imputation. So by considering imputation along with identification of power of query performance over the database, we can achieve successful improvements in the query results.

***Keywords*—**Keyword Query; Database; Query Performance; Deterministic Data Imputation

I. INTRODUCTION

From the most recent years the keyword query got more attention for database on account of their straightforwardness in looking and getting the information [1],[2],[3]. The database is made up from three building blocks namely entities, attributes and attribute values. When we give a keyword query over database, we can get the answers from different entities because answer may exist in numerous entity sets so keyword query ordinarily have different conceivable answers. The user cannot get proper answers in presence of hardness of query. Such queries give extremely inferior positioning quality, so there is have to discover required data behind queries and results are situated so that high quality answers look first in the output list [4]. The results are examined to anticipate the performance. We can also enhance the query during query handling by using approximation algorithms. Creating alternate queries or rebuilding the query helps to overcome problems included in queries [5],[6]. Here we utilize the ranking robustness principle which tells that results of simple query are steady against ranking algorithm [7]. For this rule, we present data corruption (noise) by adding or erasing attribute values. This produces corrupted version of database. To check the effectiveness of query, we are calculating Structured Robustness (SR) score by comparing similarity between original result lists and corrupted result lists [8]. We utilize spearman rank relationship to discover structured robustness score of given query [9]. Moreover missing data estimations in the database [10] is taken care of by analyzing the cooperation between the inferring based method and the recovering based method. Infer the missing values in higher extent so that a lesser number of missing values are retrieved from the web. By utilizing such kind of imputation, we can essentially enhance the recall of the inferring based systems with less cost. This new approach is called as inTeractive Retrieving-Inferring data imPutation approach (TRIP)[11].The TRIP technique is also used to identify a perfect scheduling scheme in Deterministic Data Imputation (DDI) where randomness is not present in imputed data.

Remaining paper is structured as follows:
Section II describes background, section III describes proposed work, section IV describes corresponding result analysis , section V gives conclusion and future scope of this work.

II. BACKGROUND

A. *Keyword Query*

In a keyword query, the terms user enters are used literally to retrieve any document that has all or any of those terms. The terms are not programmatically changed in any way, and match exactly only on themselves. Although provide easy access to data over database, but sometimes suffer from inferior ranking quality. Some properties of such queries are like more entities contain query terms or query matches with various attributes for same query term. [12]

B. *Structured Robustness (SR)*

The concept behind the estimation of difficult query is to calculate structured robustness score of that query. The SR guideline said that there is opposite relationship between

difficulties of query and its consistency of ranking when data is corrupted. Here we take top k results of original database to create corrupted version of database [13]. We corrupt only attribute values in top k lists that mean we can add or erase values for generating noise in database. In this way we can get corrupted list. Rerank these result lists for getting top k corrupted lists. For generating top k good results we use Probabilistic Retrieval Model for Semistructured Data (PRMS) ranking algorithm [14], where each query terms is linked into appropriate field and then calculate probability of each term into that related field. These probabilities are used in ranking of the lists and considered as weight W_j (q) for attribute values whose attribute is A_j and query q, then we calculate weight as follows:

$$W_j(q) = \frac{P(q|A_j)}{\sum_{A \in DB} P(q|A)}$$

This formula is used for getting both original and corrupted top k result lists.

After getting lists, we check whether frequencies of terms are same across original and corrupted lists to compute the similarity of the answer lists. If ranked list of original DB and corrupted database is less comparable, the given query will be more hard. Spearman rank relationship is used to find out similarity between original and corrupted answer lists. It ranges somewhere around 1 and −1, where 1, −1, and 0 exhibits positive connection, negative connection , and no connection,respectively.

Spearman rank correlation is used to compute these similarity by utilizing following equation:

$$r_s = 1 - \frac{6 \sum d_i^2}{n(n^2-1)}$$

Where d is difference between two results of each list, n is number of lists

C. Approximation Algorithms

To improve effectiveness of SR score, approximation algorithms are utilized. In Query-specific Attribute values Only Approximation(QAO-Approx) we corrupt only attribute values. Another one is Static Global Stats Approximation (SGS-Approx) which utilize the frequencies of the original database to again rank the corrupted entities and re-ranking is done during corruption. We can also combine these two algorithms to advance the efficiency of difficulty prediction.

Combined (QAO and SGS) Approximation Algorithm:

Notations:

C=No.of corrupted iterations, F=Frequency , M=Metadata, R=Original result, R'=Corrupted result, L=Top-k result lists, L'= Top-k corrupted result lists

1. For iteration i from 1to C (here C=2)
2. For each result R and attribute value A in L, Corrupt attribute value A which produces corrupted version A' of A,rank L' based on F and M
3. For each term w in query Q , compute number of w in A'
4. If number of terms w are changes in A and A'
5. Add A' to R' and Add R' to L'
6. Computes similarity between the ranked answer lists L and L' using spearman rank correlation
7. Return SR score of query Q over C rounds

D. Data Imputation

Data imputation means filling missing attribute values in the database and to fill all such missing values inTeractive Retrieving-Inferring data imPutation (TRIP) method is utilized. The TRIP approach performs inferring and retrieving in successive manner which is called as optimal scheduling scheme to fill missing values in a database with rich imputation recall. The inferring means put the missing value with value existing in database [15] with the help of functional dependencies e.g. if FD is X->Y, that means Y is functionally dependent on X. So if value of Y is missing in database then we can infer it from X. The retrieving approach means if missing value is not inferred from existing values, then we can retrieve it from web. The value may be existing in web table, web lists or plain text web documents. For retrieving values from all types of web sources [16], we have to build imputation query [17],[18] which fetches relevant web pages that contains missing value. After getting all web pages, we can retrieve missing values from the pages. Just like inferring approach, we can use FD for preparing imputation query. By performing such imputation, we can get higher precision and recall. TRIP recognizes an ideal scheduling scheme in Deterministic Data Imputation (DDI) where no randomness is in imputed data and imputation is performed by using attribute dependencies which present in table. To Identify Optimal Scheme in DDI, missing value graph is built which is called as Inference dependency graph. This graph gives us all missing values that we have to retrieve. In graph node represents missing value and edges between nodes represents dependency relation between values.

Deterministic Data Imputation (DDI) algorithm:

1. Start
2. Make set O of missing values i
3. Let S be the scheduling scheme for set O, so
   $S = \langle I_0, R_1, I_1, R_2, \ldots, R_n, I_n \rangle$
4. Infer all missing values in I_i at i-th inferring step from the set O
5. Increment value of i
6. Build an inference dependency graph to fetch the retrieving missing values R_i from graph
7. Retrieve all missing values in R_i
8. Return set O with all filled values, so O'(S)= $I_i \cup R_i$

III. PROPOSED WORK

In our proposed structure the principle commitment is enhancing keyword query search over database. To perform quick and successful search, we utilize TRIP method. We also perform elimination of characters and stopwords if present in keyword query. We use WorldNet dictionary for finding semantic meaning of every word in a given query.

A. Workflow of system

The following Fig 1. Shows the flow of all conditions in system. In this diagram two databases are represented for the same one.

In flow diagram, the admin performs two operations: data imputation and data corruption. The imputation process performs first and then query effectiveness is calculated with the help of SR score. The user gives keyword query and getting SR score of that query by using spearman's rank relationship. To remove time overhead of search, SGS-approximation algorithm is used.

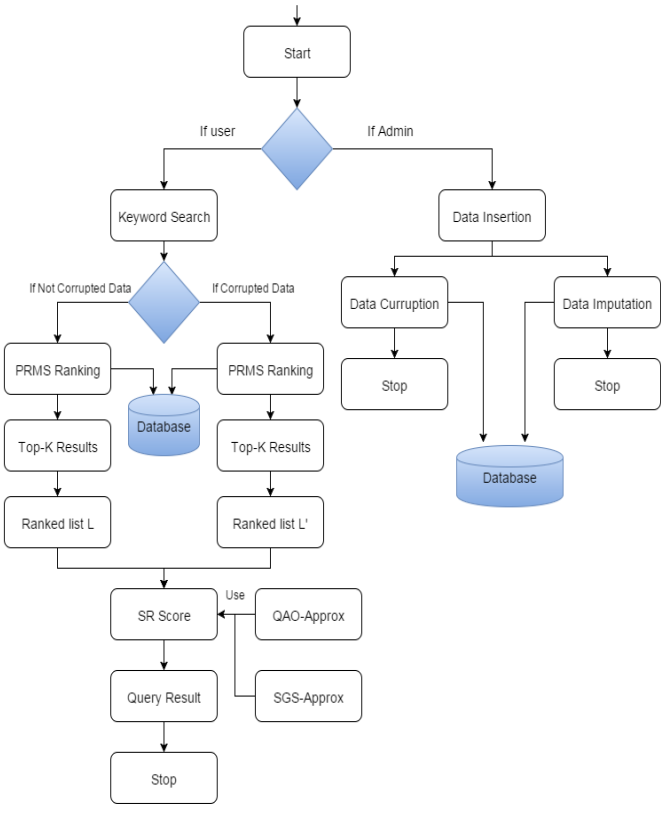

Fig 1. Flow diagram of system

IV. RESULT AND DISCUSSION

In Fig 2(a) we take 50 corrupted iterations. For each iteration we are calculating Avg. SR score within a minimum amount of time.

Iterations for Corruption	Avg. Spearman's Correlation	SR Time(corruption time)in Second
10	0.352	600
20	0.358	800
30	0.42	1000
40	0.45	1200
50	0.57	1400

Fig 2(a). Performance of Approximation Algorithm

In Fig 2(c), we are imputing missing attribute values with less time overhead. By imputing values, we are getting better search results and also identifying power of search more accurately.

Missing Ratio (%)	Time Required ($*10^3$sec)
5	3
10	10
15	15
20	18
25	26

Fig 2(b). Time required for DDI algorithm

V. CONCLUSION

The above proposed framework is able to handle the issues of database that is accurate prediction of difficult keyword query and imputing missing attribute values. The TRIP technique utilizes deterministic data imputation where missing values are imputed from the beginning. By tackling these two issues, a user can get more exact query results with higher precision.

Future work may utilize other quality conditions for imputation.

ACKNOWLEDGMENT

The authors might want to thank everyone for their master direction and made various recommendations which enhance this work.

REFERENCES

[1] N. Sarkas, S. Paparizos, and P. Tsaparas, "Structured annotations of web queries," in Proc. ACM SIGMOD Int. Conf. Manage. Data, Indianapolis, IN, USA, pp. 771–782,2010.

[2] Ganti, Y. He, and D. Xin, "Keyword++: A framework to improve keyword search over entity databases," in Proc. VLDB Endowment, Singapore, vol. 3, no. 1–2, pp. 711–722, Sept. 2010,

[3] V. Hristidis, L. Gravano, and Y. Papakonstantinou, "Efficient IRstyle keyword search over relational databases," in Proc. 29th VLDB Conf., Berlin, Germany, pp. 850–861,2003.

[4] G. Bhalotia, A. Hulgeri, C. Nakhe, S. Chakrabarti, and S. Sudarshan, "Keyword searching and browsing in databases using BANKS," in Proc. 18th ICDE, San Jose, CA, USA, pp. 431–440,2002.
[5] Nandi and H. V. Jagadish, "Assisted querying using instant response interfaces," in Proc. SIGMOD 07, Beijing, China, pp. 1156–1158.
[6] E. Demidova, P. Fankhauser, X. Zhou, and W. Nejdl, "DivQ: Diversification for keyword search over structured databases," in Proc. SIGIR' 10, Geneva, Switzerland, pp. 331–338.
[7] J. A. Aslam and V. Pavlu, "Query hardness estimation using Jensen-Shannon divergence among multiple scoring functions," in Proc. 29th ECIR, Rome, Italy, pp. 198–209,2007.
[8] Shiwen Cheng, Arash Termehchy, and Vagelis Hristidis, "Efficient Prediction of Difficult Keyword Queries over Databases", vol. 26, no. 6, June 2014.
[9] J. D. Gibbons and S. Chakraborty, Nonparametric Statistical Inference. New York, NY: Marcel Dekker, 1992.
[10] G. Batista and M. Monard. An analysis of four missing data treatment methods for supervised learning. Applied Artificial Intelligence, 17(5-6):519–533, 2003
[11] Zhixu Li, Lu Qin, Hong Cheng, Xiangliang Zhang, and Xiaofang Zhou, "TRIP: An Interactive Retrieving-Inferring Data Imputation Approach," IEEE Transaction 2015.
[12] E. Yom-Tov, S. Fine, D. Carmel, and A. Darlow,"Learning to estimate query difficulty: Including applications to missing content detection and distributed information retrieval," in Proc. 28th Annu. Int. ACM SIGIR Conf. Research Development Information Retrieval, Salvador, Brazil, pp. 512–519,2005.
[13] Y. Zhou and B. Croft, "Ranking robustness: A novel framework to predict query performance," in Proc. 15th ACM Int. CIKM, Geneva, Switzerland, pp. 567–574,2006.
[14] J. Kim, X. Xue, and B. Croft, A probabilistic retrieval model for semistructured data, in Proc. ECIR, Tolouse, France, pp. 228239, 2009.
[15] J.-J. Shen, C.-C. Chang, and Y.-C. Li. Combined association rules for dealing with missing values. Journal of Information Science, 33(4):468–480, 2007.
[16] Z. Li, M. A. Sharaf, L. Sitbon, S. Sadiq, M. Indulska, and X. Zhou. Webput: Efficient web-based data imputation. In WISE, pages 243–256, 2012.
[17] S. Brin. Extracting patterns and relations from the world wide web. The World Wide Web and Databases, pages 172–183, 1999.
[18] Z. Li, M. A. Sharaf, L. Sitbon, X. Du, and X. Zhou. Core: A context-aware relation extraction method for relation completion. IEEE Transactions on Knowledge and Data Engineering, page 1, 2013.

AUTHORS PROFILE

Priya Pujari is a M.E. Student in the Computer Engineering Department from Dr. D.Y.Patil School of Engineering and Technology at Savitribai Phule Pune University. She received the B.E. degree in Computer Science and Engineering from Kolhapur University, India. Her current research interests include information retrieval and data mining.

Prof. Arti Waghmare is working as an Assistant Professor in the Department of Computer Engineering from D.Y.Patil School of Engineering and Te- chnology which is affiliated to Saitribai Phule Pune University.She received her B.E. and M.E.degree in Computer Engineering from University of Mumbai, India.Her main areas of research interest are Information Storage and Retrieval.