International Journal of Managing Public Sector Information and Communication Technologies (IJMPICT)
Vol. 7, No. 2, June 2016

ENHANCING KEYWORD QUERY RESULTS OVER DATABASE FOR IMPROVING USER SATISFACTION

Priya Pujari[1] and Roshani Ade[2]

[1] Department of Computer Engineering, Dr. D.Y. Patil School of Engineering & Technology, Savitribai Phule Pune University, Pune, India
[2] Department of Computer Engineering, Dr. D.Y. Patil School of Engineering & Technology, Savitribai Phule Pune University, Pune, India

ABSTRACT

Storing data in relational databases is widely increasing to support keyword queries but search results does not gives effective answers to keyword query and hence it is inflexible from user perspective. It would be helpful to recognize such type of queries which gives results with low ranking. Here we estimate prediction of query performance to find out effectiveness of a search performed in response to query and features of such hard queries is studied by taking into account contents of the database and result list. One relevant problem of database is the presence of missing data and it can be handled by imputation. Here an inTeractive Retrieving-Inferring data imputation method (TRIP) is used which achieves retrieving and inferring alternately to fill the missing attribute values in the database. So by considering both the prediction of hard queries and imputation over the database, we can get better keyword search results.

KEYWORDS

Structured Data, Robustness, Data Imputation, Attribute Dependencies.

1. INTRODUCTION

From the last few years the importance of keyword query for database is increased because of their simplicity in looking and accessing the data [1] [2]. The databases contain entities, attributes and attribute values. When we fire keyword query, we can get the answer from various entities because keyword may exist in multiple entity sets so keyword queries normally have various possible answers. The user cannot get appropriate answers because of hardness of keyword query. e.g. users do not give expected information behind the query. Such queries provide very pure ranking quality.so there is need to recognize required information behind queries and results are positioned so that the expected answers looks first in list. The retrieved results are analyzed to predict the effectiveness of queries. It is possible to optimize the query during query processing. Developing alternative queries or reproducing the query helps to overcome the difficulty involved in queries. The individualities of difficult queries over databases are analyzed and suggest a technique to search such queries [3]. Here we use the Ranking Robustness Principle for databases which tells that results of easy query are stable against ranking algorithm. For this principle, we introduce data corruption (noise) by adding or deleting attribute values to check the robustness of query over original and corrupted database. We use spearman rank correlation to calculate structured robustness score. Additionally missing attribute values in the database is handled by examining the interaction between the inferring-based approaches and the retrieving-based approaches. Inferring the missing values as much as possible so that lesser number of missing values are retrieved from the web. By using such type of data imputation we can significantly improve the imputation recall of the inferring-based techniques at the least cost. This new approach is called as in Teractive Retrieving-Inferring data

im Putation approach (TRIP) to fill the missing attribute values in database[4] . The TRIP method is capable to detect an ideal inferring- retrieving scheduling scheme in Deterministic Data Imputation (DDI) where no randomness is present in the imputed data.

In the reminder of this paper, section 2 presents background and related mathematics. Section 3 describes the literature survey. In section 4, the architecture and algorithms of proposed system is described. Section 5 describes the results and lastly section 6 concludes paper.

2. BACKGROUND

2.1. Properties of Hard Query

Keyword queries on database provide easy access to data, but gives very pure quality of ranking.

Certain properties of such difficult keyword query are as follows:

- For given query, more entities contain query terms.
- Query matches with different attributes for same keyword term.
- Query matches with entities from more entity sets.

2.2. Ranking Technique

Ranking technique is used to retrieve top-K entities of query results. Here ranking algorithm called Probabilistic Retrieval Model for Semistructured Data (PRMS)[5] is used. Each query word is mapped into related field. This mapping probability is considered as a weight and gives contribution to the ranking score for both original and corrupted lists. It computes the keyword-specific weight $\mu_j(q)$ for attribute values whose attributes are T_j and query q by using following formula:

$$\mu_j(q)=\frac{P(q|T_j)}{\sum_{T\in DB} P(q|T)} \quad (1)$$

The ranking score of entity E for query Q is calculated by using Formula 2:

$$\begin{aligned} P(Q|E) &= \prod_{q\in Q} P(q|E) \\ &= \prod_{q\in Q} \sum_{j=1}^{n} [\mu_j(q)((1-\lambda)P(q|A_j) + \lambda P(q|T_j))] \end{aligned} \quad (2)$$

Where, γ is parameter value varying from 0.1 to 1 for getting better query performance prediction, A_j is attribute value, n is total number of attribute values in *E*.

2.3. Prediction of Hard Query

The basic idea behind the estimation of query difficulty is to measure structured robustness score of that query. The SR principle said that there is negative correlation between the difficultness of query and its robustness of ranking in the presence of noisy data.

2.3.1 Noise Generation (Corruption) in Database

Every attribute value is corrupted by three corruption stages: on the attribute value ,its attribute and its entity set. Here we focus on the noise introduced in the attribute values of the database because these values will propagate up to their attributes and entity set. We corrupt only the top-K entity results of the original data set. Re-rank these outcomes and shift them up to be the top-

K answers for the corrupted versions of DB. Here we check whether frequencies of terms are same across original and corrupted database to compute the similarity of the answer lists.
Computation of noise generation model:

Every attribute value $A_a \in$ A can model using V-dimensional multivariate distribution $X_a = (X_{a,1}, \ldots\ldots, X_{a,v})$,where $X_{a,j} \in X_a$ is a random variable that represents the frequency of term W_j in A_a and V be the number of distinct terms in database.The probability mass function for X_a is given in Equation 3.

$$f_{X_a}(\vec{x}_a) = p_r(X_{a,1}=x_{a,1}, \ldots\ldots\ldots\ldots, X_{a,V}=x_{a,v}) \quad (3)$$

$\vec{x}$= all $|A| \times V$ matrices that contain non-negative integers

The random variable $X_A=(X_1, \ldots\ldots, X_{|A|})$ models attribute value set A.The probability mass function for X_A is given in Equation 4.

$$f_{X_A}(\vec{x}) = p_r(X_1=\vec{x}_1 \ldots\ldots\ldots\ldots, X_{|A|}=\vec{x}_{|A|}) \quad (4)$$

Here no need to model the set of attributes *T* and set of entity sets *S.* we can derive X_T, X_S from X_A.Now $X_{DB}=X_A$. Let $\vec{x}$ is vector which contains frequencies of terms for query Q among the V distinct terms which is shown in Equation 5.

$$f_{X_a}(\vec{x}_a) = \prod_{\vec{x}_{a,j} \in \vec{x}_a} f_{X_{a,j}}(x_{a,j}) \quad (5)$$

Where $X_{a,j}$ tells how many times term appears in attribute values for corrupted database and $f_{X_{a,j}}(x_{a,i})$ is its probability.

2.3.2 Structured Robustness (SR) Score Computation

Structured Robustness (SR) Score measures the difficulty of a query by considering the differences between the rankings of the same query over the original and noisy (corrupted) database. We rank the candidate answers in DB and its corrupted versions DB', DB''. If ranked list of original DB and corrupted database is less similar, the given query will be more difficult. Spearman rank correlation is utilized to compute the similarity of the answer lists of both database. It ranges between 1 and −1, where 1, −1, and 0 demonstrate perfect positive correlation, perfect negative correlation, and no correlation, respectively.

Spearman rank correlation is calculated by using following formula:

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

SR score calculation is performed for query *Q* over database *DB* with given retrieval function *g* as follows:

$$SR(Q, g, DB, X_{DB}) = \mathbb{E}\{Sim(L(Q, g, DB), L(Q, g, X_{DB}))\} = \sum_{\vec{x}} Sim(L(Q,g,DB), L(Q, g, \vec{x})) f_{X_{DB}}(\vec{x}) \quad (7)$$

Where $\vec{x}$ contains all $|A| \times V$ matrices and Sim denotes the Spearman rank correlation between the ranked answer lists.

2.4. Approximation Algorithms

To enhance effectiveness of SR Algorithm, approximation algorithms is used. The first algorithm is Query-specific Attribute values Only Approximation(QAO-Approx) which corrupts only the attribute values that matches no less than one query term and hence can significantly decrease the time spent on the second and third levels of corruption(attribute and entity set respectively).The second algorithm is Static Global Stats Approximation (SGS-Approx) which use the statistics of the original database to again rank the corrupted entities and re-ranking is done during corruption. We merge these two algorithms to further improve the efficiency of query prediction.

2.5. Data Imputation

Sometimes database comes along with missing values and to fill all such missing attribute values, Imputation is used. Present imputation methodologies to non-quantitive information can be divided into two categories: (1) inferring- based and (2) retrieving-based methodologies. The inferring-based methodologies find replacements for the missing value from the entire data set. Yet, sometimes replaceable value does not found because it does not occur in the whole part of the data set. The retrieving-based methods are uses web resources like web tables and web lists for help, but fails to respond to large number of search queries. The TRIP (inTeractive Retrieving-Inferring data imPutation) approach performs retrieving and inferring alternately to fill missing attribute values in a dataset with high imputation recall. TRIP identify an optimal retrieving-inferring scheduling scheme in Deterministic Data Imputation (DDI), where only inference rules and imputation queries corresponding to attribute dependencies that exactly hold on the table will be used. To Identifying Optimal Scheme in DDI, Inference dependency graph is built to identify missing values to retrieve from the graph where dependency relationships between missing values are stored. At a retrieving step node represents missing value and dependency relation between values represents an edge between nodes.

3. LITERATURE SURVEY

In paper [6] V. Ganti et al. suggest overall framework that can increase an existing search interface by interpreting a keyword query to an arranged query.

In paper [7], Nikos Sarkas et al. learn latent organized semantics in web queries and produce Structured Annotations for them. Annotation is mapping of a query to arranged data table and its attributes. They provide fast and scalable tagging techniques for getting every conceivable annotations of a query over these tables.

Kevyn Collins-Thompsonan and Paul N. Bennett [8] introduce novel models and illustrations for approximating two significant measures of query presentation: query difficulty and expansion risk. Their effort brings collected features from preceding studies on query effort based on deviations between language representations of the query, collection and initial outcomes.

Oren Kurland et al. [9] presented a fundamental probabilistic prediction structure. By using their framework, they develop and describe various previously applied prediction methods that might appear totally different, but produce to share the similar formal basis also the framework is used to formulate novel prediction methodologies that leave behind the state-of-the-art.

In paper [10], Shiwen Cheng et al. investigate the characteristics of hard queries and suggest a new framework to compute the level of difficulty for a keyword query over a database, seeing both structure and content of the database and the query outcomes.

In paper [11], Arash Termehchy et al. introduced and describe independence of design, which catches property for Schema free query interfaces (SFQIs).The theoretical structure is used to compute the design amount independence delivered by an SFQI.

S. Zhang [12] introduces a new imputation approach called SN (Shell Neighbors) imputation. It fills missing val0ues in a given dataset by only using its left and right nearest neighbors with respect to each factor. The size of the sets of the nearest neighbors is determined with the cross validation method. But this method only deals with quantitative data which is continuous in datasets.

The novel web-based approach to the data imputation problem called Webput is presented in [13].It uses available information in an incomplete database in conjunction with the data consistency. WebPut also employs data imputation queries to automatically select the most effective imputation query for each missing value.

In paper [14], table is assembled from a few example rows by constructing new rows from unstructured lists on the web which extracts rows from lists, ranks them in the face of huge noise and irrelevance present in the data.

Mohamed Yakout et al.[15]describes the INFOGATHER system to automate information gathering responsibilities, such as augmenting entities with attribute values and searching attributes by means of web tables.

The structured information such as HTML tables, HTML lists and deep-web databases is combined and reproduced which is described in [16].

Paper [17] thinks about a novel issue, the interaction between record matching and data repairing where Wenfei Fan et al. suggests a uniform framework that combines repairing and matching procedures with less efforts, to cleared a database taking into account dependability requirements, expert information and coordinating guidelines.

4. PROPOSED WORK

In our proposed framework the main contribution is improving keyword query search results over database. To perform fast and effective keyword query search, we use an approximation algorithm along with structured robustness algorithm and to fill missing values, we use TRIP algorithm which perform imputation alternatively in a dataset. Sometimes user query includes characters, stop words and they make query results more complex. In our proposed work we eliminate the stop words and characters. We also provides WorldNet dictionary for semantic meaning of each word for given query. By doing this process quality of search results get improved. So this new framework for efficient keyword queries with Data Imputation gives proper output of keyword query to users.

4.1 System Architecture

Structured Robustness (SR) score recognize the difficulty of a query based on the differences between the rankings of the same query over the original and noisy (corrupted) versions of the same database. Effectiveness of SR Algorithm is enhanced by utilization of combined approximation algorithms called as Query-specific Attribute values Only Approximation (QAO-Approx) and Static Global Stats Approximation (SGS-Approx).This approximation algorithm improve the efficiency of robustness calculation by approximating various parts of the corruption and re-ranking process. TRIP performs retrieving and inferring alternately for filling missing

attribute values in a dataset. By considering both TRIP data imputation method and prediction of difficult keyword query with optimized SR algorithm which uses approximation algorithm, we can get better results and high efficiency. The following Figure.1 shows proposed system architecture.

4.2 Algorithms:

4.2.1 Optimized Structured Robustness Algorithm:

Notations:

N=No.of corrupted iterations,I= inverted index, M = metadata, L= Top-k result lists

Steps:-

1. For iteration i from 1 to N
2. Set I'← I , M' ←M , L' ←L
3. For each result R and attribute value A in L, Corrupt attribute value A which produces corrupted version A' of A
4. For each keywords w in Q , compute number of w in A'
5. If number of terms w varies in A' and A then update statistics which are stored in A',*M'* and *I'*
6. Add A' to R' and Add R' to L'
7. Rank L' based on I and M
8. Computes similarity between the ranked answer lists L and L' using spearman rank correlation
9. Return SR score of query Q over N rounds

4.2.1 Deterministic Data Imputation:

Notations:

$\mathcal{O}$ is set of missing values

I_i is set of missing values for inferring in the i-th inferring step

R_i is set of missing values for retrieving in identified minimum unlocking nodes

Steps:-

1. $\mathcal{O} = \mathcal{O} - I_i$, Infer all missing values in I_i from the set $\mathcal{O}$
2. Increment missing value in the set $\mathcal{O}$
3. Build an inference dependency graph
4. $\mathcal{O} = \mathcal{O} - R_i$, Retrieve all missing values in R_i
5. An imputation scheme $S = \langle I_0, R_1, I_1, R_2, \ldots, R_n, I_n \rangle$
   and $\mathcal{O}'(S) = (\cup_{0 \le i \le n} I_i) \cup (\cup_{1 \le i \le n} R_i)$, denote the set of filled values

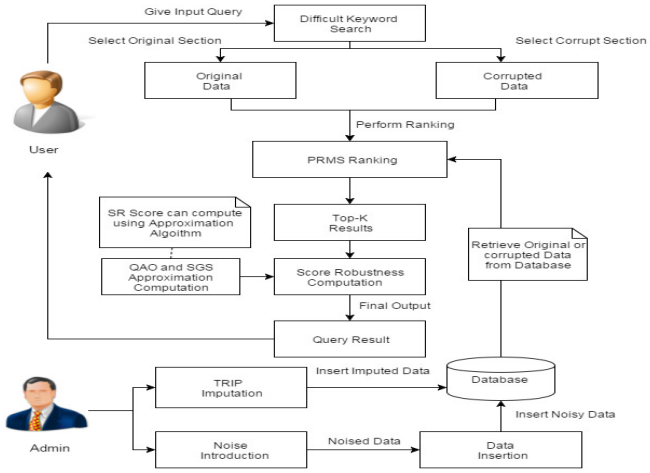

Figure 1. Proposed system architecture.

5. RESULTS

In this section, we present our experimental results.

5.1 Data Set

Here we use movies data set which is a complete table and does not contain missing values. The data set contains 300 tuples,15 attributes and hold 6 functional dependencies. To implement proposed approach, we have to manually remove some attribute values. Here we keep missing ratio up to 50%.

5.2 Accuracy

The following Table 1shows imputation accuracy of TRIP for scheduling scheme in DDI over the movie dataset by calculating the precision value for every missing ratio (%).The precision gives us a percentage of correctly imputed values from all imputed values.

Table 1. Precision of the scheduling scheme in DDI

Precision	**Missing ratio (%)**
0.7	10
0.8	20
0.9	30
1	40
1	50

6. CONCLUSIONS

The above proposed framework is capable to handle the problems of database that is prediction of difficult keyword query and missing values. The structured robustness algorithm is used to identify effectiveness of given query. If query is difficult system may give alternative query to the user. The TRIP method uses deterministic data imputation algorithm to fill missing values with low cost. By solving these two problems, a user can get more accurate keyword query search results with low errors and less time overhead.

Future work may use other attribute dependencies for inferring and retrieving based approach. Further we can apply this framework to the big data

.

ACKNOWLEDGEMENTS

The authors would like to thank all the members for their expert guidance and all others who directly or indirectly helped and made numerous suggestions which improved the quality of this work.

REFERENCES

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

[2] Y. Luo, X. Lin, W. Wang, and X. Zhou, "SPARK: Top-k keyword query in relational databases," in Proc. 2007 ACM SIGMOD, Beijing, China, pp. 115–126.

[3] Shiwen Cheng, Arash Termehchy, and Vagelis Hristidis, "Efficient Prediction of Difficult Keyword Queries over Databases", vol. 26, no. 6, June 2014.

[4] Zhixu Li, Lu Qin, Hong Cheng, Xiangliang Zhang, and Xiaofang Zhou, "TRIP: An Interactive Retrieving-Inferring Data Imputation Approach," IEEE Transaction 2015.

[5] J. Kim, X. Xue, and B. Croft, A probabilistic retrieval model for semistructured data, in Proc. ECIR, Tolouse, France, 2009, pp. 228239.

[6] V. Ganti, Y. He, and D. Xin, Keyword++: A framework to improve keyword search over entity databases, in Proc. VLDB Endowment,Singapore, vol. 3, no. 12, pp. 711722, Sept. 2010.

[7] N. Sarkas, S. Paparizos, and P. Tsaparas, Structured annotations of web queries, in Proc. ACM SIGMOD Int. Conf. Manage. Data, Indianapolis, IN, USA, pp. 771782, 2010.

[8] K. Collins-Thompson and P. N. Bennett, Predicting query performance via classification, in Proc.32nd ECIR, Milton Keynes, U.K., 2010, pp. 140152.

[9] O. Kurland, A. Shtok, S. Hummel, F. Raiber, D. Carmel, and O. Rom, Back to the roots: A probabilistic framework for query performance prediction, in Proc. 21st Int. CIKM, Maui, HI, USA, 2012, pp. 823832.

[10] S. Cheng, A. Termehchy, and V. Hristidis, Predicting the effectiveness of keyword queries on databases, in Proc. 21st ACM Int. CIKM, Maui,HI, 2012, pp. 1213-1222.

[11] A. Termehchy, M. Winslett, and Y. Chodpathumwan, How schema independent are schema free query interfaces? in Proc. IEEE 27th ICDE, Hannover, Germany, 2011, pp.649660.

[12] S. Zhang,"Shell-neighbor method and its application in missing data imputation," Applied Intelligence, 35(1):123–133, 2011.

[13] 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

[14] R. Gupta and S. Sarawagi,"Answering table augmentation queries from unstructured lists on the web," PVLDB, 2(1):289–300, 2009.

[15] M. Yakout, K. Ganjam, K. Chakrabarti, and S. Chaudhuri," Infogather: entity augmentation and attribute discovery by holistic matching w ith web tables", In SIGMOD, pages 97–108, 2012.

[16] M. Cafarella, A. Halevy, and N. Khoussainova ," Data integration for the relational web" ,PVLDB, 2(1):1090–1101, 2009

[17] W. Fan, J. Li, S. Ma, N. Tang, and W. Yu," Interaction between record matching and data repairing," In SIGMOD, pages 469–480, 2011