Providing Ranked Relevant Results for Web Database Queries

Ullas Nambiar
Dept of CS & Engg
Arizona State University
Tempe, AZ, USA
mallu@asu.edu

Subbarao Kambhampati
Dept of CS & Engg
Arizona State University
Tempe, AZ, USA
rao@asu.edu

ABSTRACT

Often Web database users experience difficulty in articulating their needs using a precise query. Providing ranked set of possible answers would benefit such users. We propose to provide ranked answers to user queries by identifying a set of queries from the query log whose answers are relevant to the given user query. The relevance detection is done using a domain and end-user independent content similarity estimation technique.

Categories and Subject Descriptors

H.2.4 [**Systems**]: Relational databases; H.3.3 [**Information Storage and Retrieval**]: Query formulation, Retrieval models

General Terms

Management

Keywords

content similarity, query suggestion, web-enabled database

1. INTRODUCTION

The rapid expansion of the World Wide Web has made a variety of databases like bibliographies, scientific databases, travel reservation systems and vendor databases accessible to a large number of lay external users. Most Web database systems are derived from a legacy database system supporting relational data model over which a form-based interface is setup for users to interact with the database. These interfaces, although easy to use, come at a price: reduced expressibility of the queries, allowing only conjunctive queries imposing strict constraints (mostly equality) to be issued over the database. Often users may not know how to precisely express their needs and may formulate queries that lead to unsatisfactory results. Such users when presented with a ranked set of results will know which subset is of interest to them.

Current solutions for providing ranked relevant answers to a query require users to provide distance metrics and importance measures for various attributes and also necessitate architectural changes to the underlying database. Given that Web databases are autonomous and have a large percentage of 'lay' users, existing solutions are not applicable to Web databases. Moreover to extract answers from Web databases one must issue a query over the database. Hence identifying relevant answers to a query necessitate extraction of new tuples from the database by issuing additional queries. Therefore we propose a domain-independent solution to provide ranked relevant answers to a user query by identifying a set of queries that are relevant to the given query. The difficulty here is two-fold. First is identifying queries that have answers relevant to the given user query. Second is to the rank the tuples according to their relevance to the user query. We begin by assuming a *query log* containing queries issued over the database is available. Given the query log, our strategy is to map a given query to a subset of relevant queries in the query log. The relevance between queries is estimated as the similarity shown by their answer tuples. A domain and end-user independent similarity estimation technique based on existing IR techniques is used to derive the similarity between answers of queries. Depending on the estimated similarities a ranked set of relevant queries are suggested to the user. All answers generated by a relevant query are considered equally relevant and inherit the similarity value of the generating query. The results are then ranked using the associated similarity values.

WWW2004, May 17–22, 2004, New York, New York, USA.
ACM 1-58113-912-8/04/0005.

2. RELATED WORK

Early approaches at providing ranked answers to queries were based on theory of fuzzy sets. But fuzzy information systems [6] require attribute values to be fuzzy in nature thereby allowing their retrieval with fuzzy query languages. The WHIRL language [3] provides approximate answers by converting the attribute values in the database to vectors of text and ranking them using the vector space model. In [8], Motro modifies a legacy database system by introducing a *similar-to* operator that uses distances metrics over attribute values to provide ranked results . The metrics required by the similar-to operator must be provided upfront by database designers. Binderberger [7] has investigated methods to extend database systems to support similarity search and query refinement over arbitrary abstract data types. Again the similarity metrics to be used to compare various data types must be provided by the users of the system. In [4], the authors propose to provide ranked answers to queries over Web databases but require users to provide additional guidance in deciding the similarity. To use the proposed system users must identify objects in the database that are closest to the objects they seek.

3. IDENTIFYING RELEVANT QUERIES

Under the relational data model, two tuples are similar only if they show same values for all the attributes i.e. exactly match each other. Hence estimating similarity between queries by looking at similarity of their tuples would show only queries having common tuples as being similar. But two queries can be considered similar even if their tuples only match partially i.e. if they have common values for some subset of the attributes. E.g., let "Author=Ullman" and "Author=Widom" be two queries on the relation Publications. The author names show no similarity, yet the authors may have publications that fall under the same Subject or appear in the same Conference or Year or a combination of all these. We believe moving from the relational model to a vector space model will help in better capturing the partial match between answers of queries.

Author=Ullman	
Co-author	C. Li:5, R. Motwani:7, ..
Title	mining:3, optimizing:5, ..
Subject	integration:5, learning:2, ..
Conference	SIGMOD:5, VLDB:5, ..
Year	2000:6, 1999:5, ..

Table 1: Supertuple for query Author="Ullman"

Query 1	**Title="web-based learning"**
Related Queries	Title="e Learning"
	Title="Web Technology"
	Conference="WISE"
Query 2	**Title="Information Extraction"**
Related Queries	Title="information filtering"
	Title="Text Mining"
	Title="Relevance Feedback"
Query 3	**Author="Abiteboul"**
Related Queries	Author="vianu"
	Author ="Dan Suciu"
	Author="Rakesh Agarwal"

Table 2: Relevant queries for three user queries

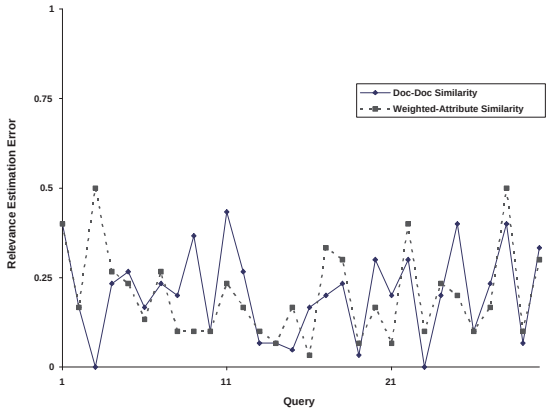

Figure 1: Error in Top-10 Estimation

Therefore we represent query results as a document of keywords thereby moving from the relational model to a vector space model.

We convert the resultset for each query to a structure called *supertuple*. The supertuple contains a bag of keywords for each attribute in the relation not bound by the query. Table 1 shows the supertuple for the query $Author = \text{“}Ullman\text{”}$ over the relation $Publications$ as a 2-column tabular structure. The similarity between two supertuples is used as the similarity between their corresponding queries. We use two similarity measures based on the *Jaccard Similarity metric* [2, 5] to estimate similarity between supertuples. *Doc-Doc similarity* and *Weighted-Attribute similarity* are two similarity measures we use to estimate the supertuple similarities. The two measures differ in the amount of structural information retained by the supertuple. To compute the Doc-Doc similarity measure, we represent the supertuple as a single bag containing all weighted keywords appearing the answerset of the query, whereas a bag of keywords for each attribute is maintained to measure the Weighted-Attribute similarity between queries.

4. EXPERIMENTS

To evaluate the effectiveness of our approach in identifying relevant answers to queries, we set up a prototype database system that extends *BibFinder* [9, 1]. *BibFinder* is a publicly-available Web data integration system, projecting a unified schema over multiple bibliography databases. BibFinder provides a form-based interface for accepting queries over the relation

$$Publications(Author, Title, Conference, Journal, Year)$$

We used 10000 queries from BibFinder's query log in our prototype system. Next we asked 3 graduate students, who are frequent users of BibFinder to evaluate the relevance of the queries we suggest. Each user was asked to pick 30 queries of their choice. For each query issued by the user, he/she had to determine how many among the suggested $top - 10$ queries they considered relevant. Table 2 presents a sample set of relevant queries recommended for three user given queries. Figure 1 illustrates the error in estimating the $top - 10$ queries relevant to a user query. Both doc-doc and weighted-average ranking measures show less than 25% average loss of precision.

5. DISCUSSION

Our current work focuses only on providing ranked answers for queries over a single database relation. Developing approaches for join queries over multiple relations is a future direction of this work. Further our approach requires the initial user query to have a non-zero resultset and assumes the query is present in the query log. We plan to extend our approach to answer queries that do not satisfy the above two constraints. A possible solution is to use co-occurrence analysis and scalar clustering techniques to identify terms that are related to terms appearing in the user query. Then we can identify another query Q' from the query log whose terms are closest to the user query Q. The queries relevant to Q' can then be used to find queries related to Q.

6. SUMMARY

We introduce a new approach for providing ranked relevant results for queries over a Web database. We use an information retrieval based approach to find the similarity among queries and use it to identify relevant answers to a given user query. To evaluate the effectiveness of our approach, we performed experiments over a real Web database system, BibFinder. The experiments indicate that the proposed approach is able to provide relevant answers with high levels of user satisfaction. The approach can be (and has been) implemented without affecting the internals of a database thereby showing that it could be easily implemented over any existing Web databases.

7. REFERENCES

[1] BibFinder :-http://kilimanjaro.eas.asu.edu/.

[2] R. Baeza-Yates and B. Ribiero-Neto. Modern Information Retrieval. Addison Wesley Longman Publishing, 1999.

[3] W. Cohen. Integration of heterogeneous databases without common domains using queries based on textual similarity. Proc. of SIGMOD, pages 201–212, June 1998.

[4] R. Goldman, N .Shivakumar, S. Venkatasubramanian, and H. Garcia-Molina. Proximity search in databases. VLDB, 1998.

[5] T. Haveliwala, A. Gionis, D. Klein, and P Indyk. Evaluating strategies for similarity search on the web. Proceedings of WWW, Hawai, USA, May 2002.

[6] J.M. Morrissey. Imprecise information and uncertainty in information systems. ACM Transactions on Information Systems, 8:159–180, April 1990.

[7] M. Ortega-Binderberger. Integrating Similarity Based Retrieval and Query Refinement in Databases. PhD thesis, UIUC, 2002.

[8] A. Motro. Vague: A user interface to relational databases that permits vague queries. ACM Transactions on Office Information Systems, 6(3):187–214, 1998.

[9] Z. Nie, S. Kambhampati, and T. Hernandez. BibFinder/StatMiner: Effectively Mining and Using Coverage and Overlap Statistics in Data Integration. VLDB, 2003.