

Structuring Keyword-Based Queries for Web Databases*

Rodrigo C. Vieira Pavel Calado[†] Altigran S. da Silva
Alberto H. F. Laender Berthier A. Ribeiro-Neto

Department of Computer Science
Federal University of Minas Gerais
31270-901 Belo Horizonte, MG, Brazil

{rcvieira,pavel,alti,laender,berthier}@dcc.ufmg.br

ABSTRACT

This paper describes a framework, based on Bayesian belief networks, for querying Web databases using keywords only. According to this framework, the user inputs a query through a simple search-box interface. From the input query, one or more plausible structured queries are derived and submitted to Web databases. The results are then retrieved and presented to the user as ranked answers. To evaluate our framework, an experiment using 38 example queries was carried out. We found out that 97% of the time, one of the top three resulting structured queries is the proper one. Further, when the user selects one of these three top queries for processing, the ranked answers present average precision figures of 92%.

Keywords: Bayesian networks, Web Queries, Web Databases

©ACM, 2002. This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in, *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries, (June 2002)* <http://doi.acm.org/10.1145/544220.544236>

1. INTRODUCTION

On-line information systems, such as digital libraries and on-line stores, have become widespread and provide access to a multitude of databases via the Web. In most cases, this access is accomplished by means of customized interfaces that use forms, navigation menus, and other browsing mechanisms. This has two important shortcomings. From the point of view of Web users, these interfaces might become too complicated for complex databases where, for in-

*Work is partially supported by the SIAM Project (MCT/CNPq/PRONEX 76.97.1016.00) and by the I3DL Project (MCT/CNPq/ProTeM-CC 680154/01-9).

[†]Supported by FCT scholarship SFRH/BD/4662/2001

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

JCDL'02, July 13-17, 2002, Portland, Oregon, USA.
Copyright 2002 ACM 1-58113-513-0/02/0007 ...\$5.00.

stance, a large number of fields or even multiple forms has to be filled. From the point of view of Web developers, it increases the cost and the time needed for developing and maintaining the interface. This situation is even worse for the case of Web sites that provide access to several distinct databases (e.g., MySimon at <http://www.mysimon.com>).

In this work, we propose a framework, based on Bayesian belief networks, to query on-line Web databases by building structured queries from a set of keywords specified by the user. This means that the user needs only to fill a single search box to formulate a query. Thus, our framework is able to provide on-line information systems with (1) an interface that is simple and intuitive to Web users, and (2) the ability to query several heterogeneous databases using a single interface. As an additional advantage, such a simple interface is particularly suitable for portable devices, such as PDAs and cellular phones, whose display space is limited.

Various works on querying Web data have been proposed. In particular, attempts to make this task easier to the end user are presented in [1, 2, 3]. Our proposed framework differs from them by the use of information retrieval techniques to structure queries and to rank query answers.

Our framework uses a Bayesian network (as in [4]) to model and derive structured queries from keywords specified by the user. The structured queries are then submitted to the database and the retrieved results are presented to the user as ranked answers. Preliminary experimental results indicate that our framework is able to accurately structure user queries and return relevant answers, requiring a minimum interaction from the user.

2. QUERY FRAMEWORK

Our proposed framework consists in (1) inputting the unstructured user query, (2) building the structured queries, (3) selecting the structured query, and (4) processing the query results. Step 1 consists in receiving the unstructured query from the user, as a set of keywords. This can be easily accomplished using a simple search box interface, and is the only user interaction needed. For step 2, a small local repository of data, representative of the database to be queried, is required. By representative we mean that: (1) the repository should contain a set of values for each attribute available in the database, and (2) each set of values should be a relevant sample of the corresponding attribute domain. For instance, for a database on movies this repository would include, movie titles, actor names, director names, etc.

Adding structure to the unstructured user query consists of determining which structure better fits the local repository. If the user query contains the keywords "george",

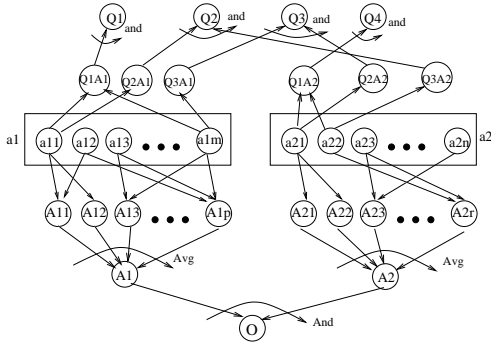


Figure 1: A Bayesian network model for two attributes to evaluate structured queries.

“star” and “wars”, our framework determines whether query $Q_1=(\text{title}=\text{“star wars” and director}=\text{“george”})$ is a better fit than query $Q_2=(\text{director}=\text{“george star” and title}=\text{“wars”})$. Queries $Q_1, Q_2 \dots Q_n$ can be built by, for instance, assigning the user keywords to each one of the available attributes. Notice that, in practice, we do not need to build all the t^k possible queries (where t is the number of attributes and k is the number of keywords) since each keyword must comply to the type of its attribute.

To determine the probability that each one of the possible queries satisfies the user needs, we model this problem as a Bayesian network, as illustrated in Figure 1. Let Q_k be the k th query, O be the database sample, A_i be the i th attribute (e.g., “director” or “title”), $Q_j A_i$ be the set of keywords of Q_k assigned to attribute A_i and a_i be the terms for attribute i . According to our Bayesian model, the derived probability is given by $P(Q_k|O) = \alpha \times P(A_1|a_1) \times \dots \times P(A_m|a_m) \times P(a_1) \times \dots \times P(a_m)$, where α is a constant, $P(A_i|a_i)$ is the probability that the query terms a_i are relevant to attribute A_i and $P(a_i)$ is a constant. $P(A_i|a_i)$ is computed as the average similarity of the query terms to each value of the set of values stored in the local repository. This similarity is computed using either the vector-space model (commonly used in information retrieval problems) for textual attributes or a normalized difference between the values, for numerical attributes.

In step 3, we have a set of plausible structured queries ranked according to $P(Q_k|O)$. We now have two alternatives to select the query to be submitted: either we select the highest ranked structured query, or ask the user to choose one among the best ranked queries. This last alternative requires an additional user interaction, but usually improves the results.

For step 4, the selected query is submitted and the returned results are ranked. The main reason for ranking the results is that, even in the case that the selected query was not completely correctly structured (say, the “title” field was incorrect, but the “director” field was correct), relevant results may still be shown. For ranking the results, a Bayesian network model similar to the one described above is used to compute the probability of a given result fitting the structured query.

3. EXPERIMENTAL RESULTS

To evaluate our query framework, a movie database, with data extracted from the IMDb (<http://www.imdb.com>) site, was used. A sample of about 280,000 movie objects was used in the experiments. Each movie object includes the at-

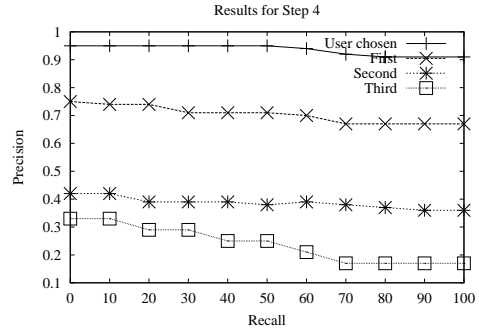


Figure 2: Results interpolated precision at 11 standard recall levels.

tributes “Title”, “Director”, “Writer”, “Year” and “Genre” and a list of “Actors”.

A set of 38 structured queries was proposed. The structure was then removed from the queries and the resultant 38 sets of keywords (unstructured queries) were submitted and processed, resulting in a ranked set of structured query alternatives for each query. Of the queries that ranked first in each set, 63% were completely correct, i.e., they correspond exactly to the original structured queries. Furthermore, 97% of the completely correct queries were always among the top three ranked queries. For those that were not completely correct, on average 69% of their attributes were correctly assigned.

With respect to the answers returned, using the highest ranked structured queries, an average of 69% correct results were obtained, as compared with the original structured queries. Due to the overlapping of attribute domains present by movie objects (e.g., attributes director, writer and actors are all from the same domain) some correct queries were ranked second and third instead of first. We believe this figure could be higher in less overlapping application domains. Using the second and the third ranked queries, as expected, precision values drop. The best results were achieved by allowing the user to choose his query from the highest ranked ones. In this case, results yielded an average of 92% in precision. Figure 2 shows the average precision/recall figures for the 38 test queries, using the user chosen and the highest ranked structured query alternatives.

4. CONCLUSIONS

We have proposed a framework for structuring keyword-based queries that allows the retrieval of data from Web databases using a simple, database-independent interface. Although still preliminary, the results indicate that our approach can be used effectively in on-line systems. Future works includes using the model in multiple databases, such as, *CDs*, *Books*, *Movies*, etc., testing different similarity measures and fusing results from different queries.

5. REFERENCES

- [1] AGRAWAL S., ET AL. DBXplorer: A system for keyword-based search over relational databases. In *Intl. Conf. on Data Eng.* (2002).
- [2] CHIANG, R., ET AL. A smart web query method for semantic retrieval of web data. *Data & Knowledge Eng.* 38, 1 (2001), 63–84.
- [3] FLORESCU, D., ET AL. Integrating keyword search into XML query processing. *WWW9 / Computer Networks*

33, 1-6 (2000), 119–135.

- [4] RIBEIRO-NETO, B., AND MUNTZ, R. A belief network model for IR. In *Proc. of the 19th Annual International Conference on Research and Development in Information Retrieval* (1996), pp. 253–260.