Preference-Driven Keyword Search in Relational Databases

Kostas Stefanidis, Marina Drosou, Evaggelia Pitoura

Department of Computer Science, University of Ioannina, Greece

{kstef, mdrosou, pitoura}@cs.uoi.gr

Abstract

Keyword-based search in relational databases allows users to discover relevant information without knowing the database schema or using complicated queries. However, such searches may return an overwhelming number of results. In this paper, we propose personalizing keyword database search by allowing users to express preferences. User preferences depend on context. Given the context of a keyword query, the related preferences are selected and used to rank the results. We present an algorithm for incorporating preferences in computing the top-k most pertinent to the user results. The algorithm uses the keyword appearances in the preferences to direct the joining of relevant tuples from multiple relations. We then extend this baseline algorithm to: (i) enhance the diversity of its results and (ii) improve its performance by sharing computational steps. Finally, we report results of an evaluation of our approach along two perspectives: performance and usability.

Keyword-based search is very popular because users do not need to be aware of either a query language or the underlying structure of the data for locating the information they want. For web search, typically, users submit keyword queries, i.e. queries consisting of a set of distinct keywords, while a search engine returns the data items that encompass the given keywords.

In relational databases, existing keyword search approaches exploit the database schema to propose algorithms for answering keyword queries. The schema graph of a database is a directed graph capturing the foreign key relationships in the schema. As a reference example, consider the movie database instance shown in Fig. 1. In this database, the relations are: $Movies$, $Play$ and $Actors$ (for simplicity, M, P and A). The foreign key to primary key relationships are: $P \rightarrow M$ and $P \rightarrow A$. Assuming that a user, say Anna, submits the query q = $\{thriller, B.\ Pitt\}$, the results of q are the $thriller$ movies *Twelve Monkeys* and *Seven* both with *B. Pitt*.

Given the abundance of available information, exploring the contents of a database is a complex procedure that returns a huge volume of data. However, users would like to retrieve only a small piece of it, namely the most relevant to their interests.

Previous approaches for ranking the results of keyword search include adapting IR-style document relevance ranking strategies ([1]) and exploiting the link structure of the database ([2, 3, 4]). In this paper, we propose personalizing database keyword search by letting users indicate their personal interests through preferences. Then, the results of each query are ranked according to the interests of the user that submitted the query.

Often, user preferences vary depending on specific conditions. For example, Anna may prefer movies by *S. Spielberg* to movies by *Q. Tarantino* only when their genre is *drama*. This means, that if Anna submits the keyword query $q = \{drama\}$, ideally, she would like to receive data about *Q. Tarantino drama* movies only in case there are no, or not enough, data about *S. Spielberg drama* movies, i.e. *S. Spielberg drama* movies are ranked higher than *Q. Tarantino* drama movies.

In this paper, we extend existing keyword search methods by adding the notion of *contextual keyword preferences*. A contextual keyword preference consists of two parts: the $context$ and the $choice$. Both $context$ and $choice$ are expressed through keywords. The meaning of a contextual keyword preference is that the preference specified by $choice$ holds under a specific $context$. For example, consider the following contextual keyword preferences:

- ($\{$*drama*$\}$, *S. L. Jackson* $\succ$ *L. Neeson*)
- ($\{$*drama, S. Spielberg*$\}$, *L. Neeson* $\succ$ *S. L. Jackson*)

The first preference denotes that in *drama* movies *S. L. Jackson* is preferred to *L. Neeson*. However, in the context of *S. Spielberg drama* movies, the latter actor is considered more important. To increase expressiveness, we further consider the case of *composite contextual keyword preferences* in which choices can be specified using conjunctions of keywords.

To compute the results of a query q we employ the set of contextual keyword preferences P_C with context C equal to q. We rank these results based on the choices of P_C. Considering that the keywords appearing in these choices follow a strict partial order, we introduce the *multiple level winnow operator* to retrieve the most preferable ones. However, queries are often too specific to match any context of the available preferences. To handle this issue, we consider *contextual relaxation* as the approach of employing the preferences with context more general than the one defined by q. For example, for the query $\{$*drama, F. F. Coppola*$\}$, we may use the preference ($\{$*drama*$\}$, *S. L. Jackson* $\succ$ *L. Neeson*).

We also define the top-k results of a query q based on contextual keyword preferences. We first propose an algorithm for computing those results and then a suite of variations that exploit the overlap of the intermediate results and achieve the desirable by the users diversity of results. We evaluate our approach along two perspectives: performance and usability. Our performance experiments focus on the proposed algorithm for computing the top-k results and its various enhancements. Our usability study evaluates the overhead imposed to users for specifying contextual keyword preferences as well as their satisfaction from the quality of the results.

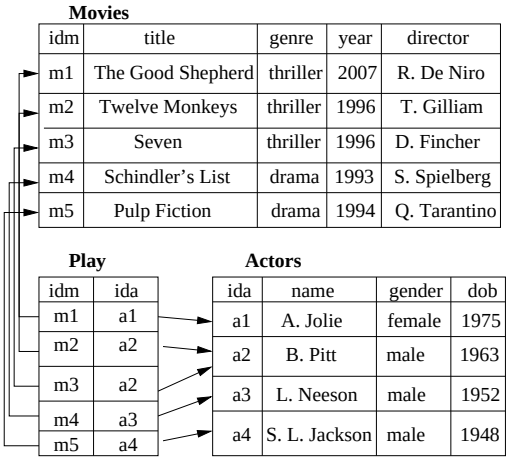

Figure 1: Database instance.

References

[1] V. Hristidis, L. Gravano, and Y. Papakonstantinou, "Efficient ir-style keyword search over relational databases," in *VLDB*, 2003, pp. 850–861.

[2] V. Hristidis and Y. Papakonstantinou, "Discover: Keyword search in relational databases," in *VLDB*, 2002, pp. 670–681.

[3] S. Agrawal, S. Chaudhuri, and G. Das, "Dbxplorer: A system for keyword-based search over relational databases," in *ICDE*, 2002, pp. 5–16.

[4] G. Bhalotia, A. Hulgeri, C. Nakhe, S. Chakrabarti, and S. Sudarshan, "Keyword searching and browsing in databases using banks," in *ICDE*, 2002, pp. 431–440.