KEYMANTIC: A KEYWORD-BASED SEARCH ENGINE USING STRUCTURAL KNOWLEDGE

Francesco Guerra
Dipartimento di Economia Aziendale, Università di Modena e Reggio Emilia, Italy

Sonia Bergamaschi, Mirko Orsini, Antonio Sala
Dipartimento di Ingegneria dell'Informazione, Università di Modena e Reggio Emilia, Italy

Claudio Sartori
Dipartimento di Elettronica, Informatica e Sistemistica, Università di Bologna, Italy

Keywords: Keyword-based search engine, Database, semantics, Metadata, Querying process.

Abstract: Traditional techniques for query formulation need the knowledge of the database contents, i.e. which data are stored in the data source and how they are represented.
In this paper, we discuss the development of a keyword-based search engine for structured data sources. The idea is to couple the ease of use and flexibility of keyword-based search with metadata extracted from data schemata and extensional knowledge which constitute a semantic network of knowledge. Translating keywords into SQL statements, we will develop a search engine that is effective, semantic-based, and applicable also when instance are not continuously available, such as in integrated data sources or in data sources extracted from the deep web.

1 INTRODUCTION

Querying structured data sources addresses several critical issues, especially in large and complex sources that may not easily be known and managed by users. The query formulation needs the knowledge of the database contents, i.e. which data are stored in the data source and how they are represented. Let us consider, for example, relational databases, where expressing a query means to be able to select the right tables and attributes of a database, and to specify proper constraints on attributes. Such a process implies overcoming some critical tasks concerning structural and lexical aspects:

1. the user selects the tables and attributes of interest on the basis of their names, which may be misleading or not meaningful (lexical aspect);
2. the user expresses constraints on attribute without having accurate knowledge of the domain. Thus, s/he may define too selective or, vice-versa, too broad or illegal selection clauses (lexical aspect);
3. the user does not know the relationships between the tables and, consequently, it is hard to pose multi-table queries (structural and lexical aspect).

Therefore, there is a direct connection between the user's ability to express queries and the knowledge of what and how the data are stored. If the user knows the database contents, the query process requires only the translation of the user's search criteria into a proper formalism. On the contrary, selective queries require an analysis of the data source, i.e. a complex and time consuming task, for users that do not know the database contents. This issue is very significant when you deal with real applications, where data structures and instances are heterogeneous, since they come from different and unknown sources. The same issue has to be addressed in querying an integrated data source, i.e. the unified view that results from the application of an integration methodology to a set of data sources (Lenzerini, 2002). In these cases, the same real world objects are represented in different sources with different data schemata, names of

Guerra F., Bergamaschi S., Orsini M., Sala A. and Sartori C. (2009).
KEYMANTIC: A KEYWORD-BASED SEARCH ENGINE USING STRUCTURAL KNOWLEDGE.
In *Proceedings of the 11th International Conference on Enterprise Information Systems - Databases and Information Systems Integration*, pages 241-246
DOI: 10.5220/0002155802410246

attributes and domains of values, and, consequently, synonym, polysemic, broader terms have to be taken into account in query formulations. The knowledge of the data collected in the "integrated" view is mandatory for expressing selective queries.

The research community has developed techniques for querying structured data sources that, from the user's perspective, can be roughly grouped into two categories:

1. query engines where the structures of the sources allow the users formulating selective queries by means of a specific query language;
2. keyword-based search engines, which exploit information retrieval techniques for selecting the instances closer to the terms provided by the users.

Query engines allow users to express complex queries with selection clauses defining constraints on the results. On the other hand, a user has to know the structure of the sources (i.e. names of the tables, the names and domains of attributes, the relationships between the tables) and a query language for writing effective queries. The research community has been involved in developing tools for supporting users in writing queries (according to the query by example approach (Zloof, 1975)) and for visualizing data sources structures (see (Katifori et al., 2007) for a survey).

Keyword-based search engines are more intuitive for the user, but they support less selective queries, since they detect instances in data sources that satisfy specified keywords. Some effective keyword-based search techniques applied to relational databases have been proposed (see section 2 for some related work). All those systems apply information retrieval techniques to the database instances, and, consequently, they suffer of several limitations. First, they are based on instance-analysis. This is a critical aspect, because it limits their action area to materialized data sources. Thus, traditional keyword-based search engine cannot be applied to integrated data sources, or to data sources which are part of the deep web. Second, they do not take into account the particular knowledge that is conveyed by database structures for the search purposes. Current techniques exploit database information mainly to identify the same instances in different tables (typically by means of foreign keys).

We claim that the current approaches for keyword-based searching on structured data sources may be improved in two directions. Firstly, the search process may be coupled with techniques derived from database systems and information retrieval. This is a new research direction, with challenging perspectives (Weikum, 2007). Secondly, we think that techniques based on semantics may improve the development of an effective keyword-based search. There is a lot of research on data and semantics. In data integration, techniques based on semantics are exploited for making the integration process as automatic as possible (Doan and Halevy, 2005). The Semantic Web aims at building a web of data, where semantic techniques are exploited for allowing data to be shared and reused across application, enterprise, and community boundaries[1]. Some researchers suggest the application of semantic web techniques to deep web, for improving the search (Wright, 2008).

In this paper, we discuss the development of a keyword-based search engine for structured data sources that exploits the semantics associated to the data structures for improving the results, by means of the exploitation of a *semantic network of knowledge*. In particular, we claim that semantics may be added to the process by taking into account:

1. **the semantics associated to the data schemata** may be used for improving searches. In particular, let us consider a relational database: semantic relationships join the table with the corresponding attributes, other relationships connect the attributes belonging to the same table, foreign keys link tables with each other, attribute domains allow addressing the search to specific attributes. Such semantics constitute a network of relationships that has to be exploited for selecting the tables containing the user's keywords;
2. **lexical knowledge** may be used for two purposes: first to analyze the keywords inserted by the user in order to "fit in" them with the lexicon used in the sources: a set of functions for translating a term in a set of synonym, similar, broader/narrower, meronym terms may turn a keyword into a set of keywords with higher recall. Obviously, such operation has to be properly parametrized since it may decrease the result precision. Second, lexical knowledge allows defining relationships connecting the database schema elements, enhancing the semantic network of knowledge;
3. **new kinds of metadata** may be defined to improve searches. Statistical indexes based on instance analysis may support the search process by indicating the data structures where to address the research. Keywords introduced in previous searches and the corresponding obtained results may be exploited to build and update indexes, with the goal of improving next searches and ranking the results. Besides these kinds of metadata, we think that it may be useful to intro-

[1] http://www.w3.org/2001/sw/

duce a "semantic" metadata, i.e. a metadata that "synthesizes" the knowledge held by the instances represented in tables/attributes. For this reason, we think to exploit and extend "relevant values" as defined in (Bergamaschi et al., 2007). Relevant values, which are automatically computed, represent an attribute domain with a reduced list of its most important values w.r.t. a semantics based on lexical and syntactic knowledge. Relevant values may be used to address the search to specific tables and attributes that have their relevant values "similar" to the user provided keywords.

4. **external tools and external ontologies** may be used to build and refine the semantic network. In particular, WordNet[2] may be exploited for deriving lexical knowledge, ontologies and taxonomies available in Internet (e.g.: SUMO[3], OpenCyc[4], dmoz[5]) for deriving new semantic relationships.

By taking into account these semantics, we aim at developing a keyword-based search engine working even in absence of knowledge about instances. Our proposal, namely Keymantic, conceives the search engine as a component for query routing, i.e. works coupled to a generic relational database management system, with the task of identifying the relevant domain(s) of a query and then mapping the keywords into a query to the fields of the data schema for that domain (Madhavan et al., 2006).

Our proposal extends the issues for the implementation of a keyword search engine based on query routing introduced in (Madhavan et al., 2007), according with the following outline: section 2 introduces some related work, section 3 describes the functional architecture of our proposal, and the main features of the modules that constitute it and finally section 4 sketches out some conclusion.

2 RELATED WORK

Works related to the issues discussed in this paper are in the areas of Semantic Web, matching based on semantic techniques and keyword-based search engines.

The first two topics have been extensively investigated in the literature: a complete overview of these themes is out of the purposes of this paper. Besides the references inserted in the text, we would like to highlight that current main challenges for creating a "web of data", i.e. the semantic web purpose, concern the application of semantic techniques to the deep web, thus building a semantic deep web (Wright, 2008) and the application of techniques developed for DBMS to the structured data that exist on the Web today (Madhavan et al., 2006).

This paper starts from the assessments introduced in (Guha et al., 2003), where the concept of semantic search is defined as *navigational search*, i.e. when the user provides keywords that s/he expects to find in the data, and *research search*, i.e. the user provides a phrase that is intended to denote the object an user wants to have information about. We take into account and extend the ideas for the implementation of a keyword search engine based on query routing introduced in (Madhavan et al., 2007), where a new data integration architecture with features similar to the ones depicted in this paper, PAYGO, is proposed for web scale data integration. Few other proposals may be compared to Keymantic: EasyQuery (Li et al., 2007), that follows an approach based on statistic and syntactic matching for query routing, and the YACOB system (Sattler et al., 2005), that does not follow a semantic approach and is mediator based data integration system oriented.

Finally, Keymantic differs from the approaches adopted by the current keyword-based search engines for relational databases under development by the research community (e.g.: BANKS (Aditya et al., 2002), DISCOVER (Hristidis and Papakonstantinou, 2002), DBXplorer (Agrawal et al., 2002), Prècis (Simitsis et al., 2008)). All these systems do not really take into account intensional knowledge extracted from database schemata, but they apply and extend information retrieval techniques to their instances. Challenges for such systems are mainly related to query optimization and ranking (see (Liu et al., 2006)), keyword search on multiple data sources (see (Sayyadian et al., 2007)), identification of related records in different tables - with the application of join techniques or other techniques (BANKS, DBXplorer, DISCOVER and (Yu et al., 2007)), and the development of a new search paradigm (Prècis). On the contrary, our proposal aims at exploiting the structural knowledge available in the data sources in conjunction with the extensional knowledge, and foresees the definition of easy languages for expressing selection clauses.

[2] http://wordnet.princeton.edu
[3] http://www.ontologyportal.org/
[4] http://www.opencyc.org/
[5] http://www.dmoz.org/

3 FUNCTIONAL ARCHITECTURE OF KEYMANTIC

From the architectural point of view, the search engine is designed to be an add-on to allow querying generic relational databases. Figure 1 shows that Keymantic may functionally be divided into four modules, with specific tasks. The *pre-processing module* builds the semantic network, stored in a Knowledge base repository, that is exploited by the *searching module* to select the tables and attributes that collect the required data. The *keyword analysis module* is in charge of analyzing user's input and, by means of the knowledge held in the semantic network, transforms it into a corresponding set of terms closer to the domains of the involved database. Finally, the *post-processing module* aims at providing the results to the user, cleaning them from duplicated items and ranking them.

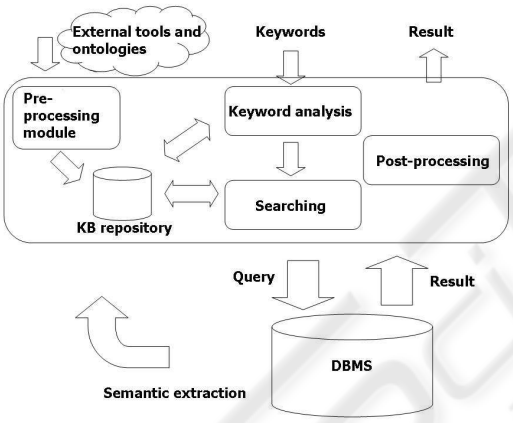

Figure 1: The functional architecture of Keymantic.

Next sections adds some details on the the main features and issues of each module.

3.1 The Pre-processing Module

The pre-processing module is in charge of the creation of indexes and data structures to be exploited for the keyword analysis and the search task. Our approach exploits the following elements:

1. The semantic network: a set of relationships that connect the elements of the data structures of the involved database. The relationships generate a weighted path that connects tables and attributes. The relationships are generated by taking into account structural and lexical knowledge extracted from the sources (see (Beneventano et al., 2001; Bergamaschi et al., 2001) for an example);
2. Attribute domain evaluation: indexes storing information about the attribute domains may be exploited for checking the compatibility between keywords and data types of the attributes;
3. "Relevant values" (Bergamaschi et al., 2007): since they are automatically computed values of an attribute that allow synthesizing the domain of an attribute, they may be exploited, by means of specific matching techniques, for addressing the keyword search on the most promising attribute.

3.2 The Keyword Analysis Module

A keyword-based search engine foresees as input a set of keywords and provides as a result the instances of the data sources containing those keywords. The analysis of the users input may allow distinguishing *schema-related keywords* (that indicate on which portion of the schema the search should be addressed) from *intensional related keywords* that allow filtering out the results. Such an analysis is mainly based on matching techniques (Giunchiglia et al., 2007) that compute the proximity of every keyword with respect to the terms used to name the elements of the data sources and which are collected in the semantic network by the pre-processing module.

Some functions may be developed and applied to the keywords in order to enrich the search process. We divide such functions in two categories: *conceptualization* and *transformation functions*. In our experience, most of the keyword a user provide are about instances. The metadata describing the database structures refers to abstract concepts. Thus, we need a conceptualization function, which transforms data in metadata, for associating a keyword to the most promising database structure. We think that only the semantics extracted from external knowledge sources may support this task. In particular, it is possible to exploit the "instance" function provided by WordNet that returns the concept associated to a term. Some other conceptualization functions may be developed by taking into account Wikipedia[6] and Dmoz. For each term collected, Wikipedia indicates a set of categories where the term belongs. Dmoz organizes information in nested categories. For each term, it is therefore possible to have a list of terms that represent increasingly broader conceptualizations. Lexical transformation are based on WordNet. For each element, WordNet returns synonym, broader/narrower, meronym terms thus allowing a richer search, decreasing at the same time the precision level of the result. This aspect has to be taken into account in ranking the results.

[6] http://en.wikipedia.org/

Such a process may be further refined by providing an annotation of the keywords with respect to a lexical reference or an ontology. By associating a definite meaning to the keywords, the recall of the results improves. Some disambiguation techniques may be applied to automatize the process. Such techniques are typically based on context. Since the context provided by few keywords is too poor to be exploited by automatic software, a graphical interface to support the user in this task has to be developed.

Finally, the adoption of an easy language for the keyword definition has to be evaluated, exploiting the structural characteristics of the underlying databases to express simple selection predicates and approximate searches. In particular, we will evaluate the possibility of expressing keywords such as "vehicle:price=15000", stating that the search is addressed to vehicles whose price is 15000 euro. This kind of keywords allows to search for instances that have given values (15000) of a specific structural element (the attribute price of the table vehicle). The approximate searches among structural elements will provide not only results from the table "vehicle" (if it exists) but also those coming from tables whose name is semantically close to vehicle (for example the table "car",...). The approximate searches on the extensional side will provide, not only the vehicles whose price is 15000 euro, but also those with a price close to that value.

3.3 The Searching Module

This module performs two tasks: the selection of the searched tables and attributes and the rewriting of the user provided keyword into an SQL query to be executed by the DBMS holding the data.

The first operation is performed by applying matching techniques to schema-related keywords (or the ones obtained by the application of conceptualization functions). The goal of this task is to identify the most promising tables and attributes on the basis of the results of the pre-processing phase. There is a rich literature about matching techniques (see for example (Giunchiglia et al., 2007)). For our purposes, we aim at extending some algorithms for approximate matching (see (Navarro, 2001) for a survey) in order to base our work on the semantic network computed in the pre-processing phase.

The second operation transforms keywords resulting from the previous phase into SQL queries. It is a straightforward process, since the target attribute/table computed in the previous step will be translated into a select/from clause and intensional related keyword that will define the selection clause of the query. Notice that (a) each keyword may generate more than one query, according to the semantic network and to the applied transformation functions. Each query result differently ranks the user keyword; (b) a trivial approach will be adopted for reconciliating keywords that cannot be referred to the same table (or to tables that may not be connected by join operations). In this case the user will be informed of the inconsistency and asked to change the set of keywords.

3.4 Post-processing

This module concerns the analysis of the query results to be proposed to the user. We think that two tasks have to be achieved in this phase: data fusion and result rank.

Data fusion is the identification and the handling of the same real world object in different databases in order to provide the user with a unique and correct answer. Several techniques have been proposed for solving the issues related to data fusion. In particular, (Naumann et al., 2006) proposed an automatic technique that shall be adapted and extended for a keyword-based search engine.

Results will be ranked according to the keywords used for generating the SQL queries. In particular, the transformations to the keywords obtained applying the techniques introduced in section 3.2 are weighted and then exploited to rank the results.

4 CONCLUSIONS AND FUTURE WORK

In this paper, we presented our preliminary work for the development of a keyword-based search engine for data schemata. We think that research on this topic is challenging: firstly, it allows the combination of techniques from information retrieval and database systems; secondly, it investigates issues that are complementary and orthogonal to the ones addressed by current search engines; thirdly, it allows the reuse and the extension of techniques previously developed in the field of semantic web, data matching, data fusion. We think that our work may be applied in several domains, such as non-materialized integrated sources and deep web data sources.

Future work will be devoted to the development, implementation and testing of each component where a particular attention will be addressed on the optimization of the developed techniques, concerning especially the response time. The search engine will be evaluated in different application domains, such

as the TPC-H benchmark[7], which provides a relevant database for the industrial domain and it is an important reference for similar applications.

ACKNOWLEDGEMENTS

This work has been partially funded by the Italian Ministry of University and Research within the project "NeP4B - Networked Peers for Business" and by the Fondazione Cassa di Risparmio di Modena within the project "Searching a needle in amounts of data!".

REFERENCES

Aditya, B., Bhalotia, G., Chakrabarti, S., Hulgeri, A., Nakhe, C., Parag, and Sudarshan, S. (2002). Banks: Browsing and keyword searching in relational databases. In *VLDB 2002, Proceedings of 28th International Conference on Very Large Data Bases, August 20-23, 2002, Hong Kong, China*, pages 1083–1086. Morgan Kaufmann.

Agrawal, S., Chaudhuri, S., and Das, G. (2002). Dbxplorer: A system for keyword-based search over relational databases. In *ICDE*, pages 5–16. IEEE Computer Society.

Beneventano, D., Bergamaschi, S., Guerra, F., and Vincini, M. (2001). The momis approach to information integration. In *ICEIS (1)*, pages 194–198.

Bergamaschi, S., Castano, S., Vincini, M., and Beneventano, D. (2001). Semantic integration of heterogeneous information sources. *Data Knowl. Eng.*, 36(3):215–249.

Bergamaschi, S., Sartori, C., Guerra, F., and Orsini, M. (2007). Extracting relevant attribute values for improved search. *IEEE Internet Computing*, 11(5):26–35.

Chan, C. Y., Ooi, B. C., and Zhou, A., editors (2007). *Proceedings of the ACM SIGMOD International Conference on Management of Data, Beijing, China, June 12-14, 2007*. ACM.

Doan, A. and Halevy, A. Y. (2005). Semantic integration research in the database community: A brief survey. *AI Magazine*, 26(1):83–94.

Giunchiglia, F., Yatskevich, M., and Shvaiko, P. (2007). Semantic matching: Algorithms and implementation. *J. Data Semantics*, 9:1–38.

Guha, R. V., McCool, R., and Miller, E. (2003). Semantic search. In *WWW*, pages 700–709.

Hristidis, V. and Papakonstantinou, Y. (2002). Discover: Keyword search in relational databases. In *VLDB 2002, Proceedings of 28th International Conference on Very Large Data Bases, August 20-23, 2002, Hong Kong, China*, pages 670–681. Morgan Kaufmann.

Katifori, A., Halatsis, C., Lepouras, G., Vassilakis, C., and Giannopoulou, E. G. (2007). Ontology visualization methods - a survey. *ACM Comput. Surv.*, 39(4).

Lenzerini, M. (2002). Data integration: A theoretical perspective. In Popa, L., editor, *PODS*, pages 233–246. ACM.

Li, X., Meng, W., and Meng, X. (2007). Easyquerier: A keyword based interface for web database integration system. In Ramamohanarao, K., Krishna, P. R., Mohania, M. K., and Nantajeewarawat, E., editors, *DASFAA*, volume 4443 of *Lecture Notes in Computer Science*, pages 936–942. Springer.

Liu, F., Yu, C. T., Meng, W., and Chowdhury, A. (2006). Effective keyword search in relational databases. In Chaudhuri, S., Hristidis, V., and Polyzotis, N., editors, *SIGMOD Conference*, pages 563–574. ACM.

Madhavan, J., Cohen, S., Dong, X. L., Halevy, A. Y., Jeffery, S. R., Ko, D., and Yu, C. (2007). Web-scale data integration: You can afford to pay as you go. In *CIDR*, pages 342–350. www.crdrdb.org.

Madhavan, J., Halevy, A. Y., Cohen, S., Dong, X. L., Jeffery, S. R., Ko, D., and Yu, C. (2006). Structured data meets the web: A few observations. *IEEE Data Eng. Bull.*, 29(4):19–26.

Naumann, F., Bilke, A., Bleiholder, J., and Weis, M. (2006). Data fusion in three steps: Resolving schema, tuple, and value inconsistencies. *IEEE Data Eng. Bull.*, 29(2):21–31.

Navarro, G. (2001). A guided tour to approximate string matching. *ACM Comput. Surv.*, 33(1):31–88.

Sattler, K.-U., Geist, I., and Schallehn, E. (2005). Concept-based querying in mediator systems. *VLDB J.*, 14(1):97–111.

Sayyadian, M., LeKhac, H., Doan, A., and Gravano, L. (2007). Efficient keyword search across heterogeneous relational databases. In *ICDE*, pages 346–355. IEEE.

Simitsis, A., Koutrika, G., and Ioannidis, Y. E. (2008). Précis: from unstructured keywords as queries to structured databases as answers. *VLDB J.*, 17(1):117–149.

Weikum, G. (2007). Db&ir: both sides now. In (Chan et al., 2007), pages 25–30.

Wright, A. (2008). Searching the deep web. *Commun. ACM*, 51(10):14–15.

Yu, B., Li, G., Sollins, K. R., and Tung, A. K. H. (2007). Effective keyword-based selection of relational databases. In (Chan et al., 2007), pages 139–150.

Zloof, M. M. (1975). Query-by-example: the invocation and definition of tables and forms. In Kerr, D. S., editor, *VLDB*, pages 1–24. ACM.

[7] http://www.tpc.org