On Semantic Search Algorithm Optimization

Alexander Gusenkov(✉) and Naille Bukharaev(✉)

Kazan Federal University, Kazan, Russia
gusenkov.a.m@gmail.com, boukharay@gmail.com

Abstract. In the article we consider, on the example of development of a relational database (RDB) information system for Tatneft oil and gas company, an approach to organization of effective search in large arrays of heterogeneous data, satisfying the following essential requirements.

On the one hand, the data is integrated at the semantic level, i.e. the system supports the presentation of data, describing its semantic properties within an unified subject domain ontology. Accordingly, end user's request are formulated exclusively in the subject domain terminology.

On the other hand, the system generates unregulated SQL-queries, i.e. the full text of possible SQL-queries, not just values of particular parameters, predefined by the system developers.

Considered approach includes both the possibilities of increasing the reactivity of the universal SQL queries generation scheme as well as more specific optimization possibilities, arising from the particular system usage context.

Keywords: Semantic search · Intellectual search · Ontology · Algorithm optimization

1 Introduction

With all diversity of its aspects, the core nature of the search problem in heterogeneous resources is determined by the type of data integration. Kogalovsky in [1] proposed to distinguish physical, logical and semantic levels of integration.

Integration at the physical level presumes converting data from various sources into a unified physical presentation format. The logical level supports the ability to access data from various sources in common terms of some global logical scheme. The semantic level supports access to data exclusively in terms of its semantic properties, described in a subject domain ontology.

The main advantage of the semantic approach lies in its obvious proximity to the task of intellectual search in multiple data sources of various logical structure and physical organization, related to one subject domain [2, 3]. The description of the domain is considered in this case as a configurable parameter of the corporate information system.

The inevitable price of the advantages of the semantic approach is the greater complexity of its implementation, which means

Á. Rocha et al. (Eds.): WorldCIST'19 2019, AISC 930, pp. 472–481, 2019.
https://doi.org/10.1007/978-3-030-16181-1_45

- theoretical novelty of the methods used;
- structural complexity of the system architecture, in which all three levels of data integration are present;
- computational complexity of the main semantic search algorithm.

Many of the arising problems can be solved at the general theoretical level. Currently, research in the field of computer linguistics is actively developing. Significant progress has been made in the development of electronic dictionaries, thesauruses, ontologies and algorithms for automatic extraction of information from the natural language text. Within the framework of this direction, a large number of specialized search systems for various subject domains have been developed. In particular, in [4–6] the semantic approach was applied to the integration of relational databases (RDB).

In the approach described by the authors in [4–6], physical database models, logical subject domain model and thesaurus of user terminology form the basis of RDBs integration. All of these information resources are presented uniformly in the ontology formalism. To build the ontologies, natural language text extracted from the names and descriptions of the RDBs tables and their attributes was used as a source of information. The proposed approach was successfully implemented to develop Tatneft oil and gas company intellectual search system, which showed high relevance of results for overwhelming majority (over 90%) of standard user queries.

Above we've mentioned the problem of computational efficiency as inevitably arising in development of the genuine intelligent search systems, which do not restrict the end user's language to a small set of predefined parametric queries. Indeed, since the user's queries are formulated exclusively in semantic terms, then the main algorithm, which has to locate the relevant information in the RDBs, in that case is of exponential complexity.

Further we explore the ways to increase the search algorithm efficiency without losing the expressive power of the end user query language.

The content of the article is structured as follows. In Sect. 2 we analyze the main existing approaches to RDB intelligent search systems development. Section 3 provides a general description of the authors' approach to organization of the search procedure and related data structures. Section 4 contains an overview of the search system architecture. In Sect. 5 the concept of intelligent search is specified through description of the end user query language. Section 6 describes preprocessing stage as a process of pre-tuning and initial optimizing further queries execution. Section 7 proposes two approaches to the search algorithm optimization; the first of them is based on the storage of the query history and the second one utilizes specifics of the subject area.

2 Intelligent Search RDB Systems

Since the mid-1990s several techniques to make full unregulated interaction with databases affordable for the users without knowledge of the SQL language and its many DBMS specific dialects have been developed. These techniques initially were mainly centered around the idea of visual programming and they generated a large number of products for constructing queries and generating reports. Now there is a lot

of products of this type; among classic examples are Crystal Reports [7] and Oracle Discoverer, which is currently is a part of Oracle Fusion Middleware [8].

One of the most famous systems of this type is Microsoft Semantic Search [9], which is the successor of Microsoft English Query. The system is based on syntactically-oriented templates, associated with the subject domain model and, through it, with the database schema. To configure the system, one needs to specify at first the models of the database and the subject domain and then select from the list of English grammar templates an appropriate one for each database relation.

Although the report generators really allow to build any database query, attempts to make them serve as an end user tool have not been successful. These analysis and reporting instruments are aimed mainly on advanced users who should know well the database structure and have sufficiently good understanding of how SQL-queries are built.

An end user usually knows the subject domain much better than the database developers and that's why he/she may need to formulate complex queries which are not supported by the standard interface. Obviously, for such a professional, the main obstacle in such cases will be necessity to know the database definition and to be able to build complicated SQL queries. In connection with this, attempts to develop a natural language (NL) based interface for database access have repeatedly been made.

From remarkable examples of the kind for the Russian language, it is possible to single out the InBase system, developed by the school of Narinyani [10]. A distinctive feature of this approach is the usage of outrunning semantic analysis during parsing and understanding of user queries. The parsing here is based on the object model of the subject domain, linked by the designer to the database model.

Let's note that the development of such systems is often negatively affected by too straightforward understanding of what is actually a "natural language interface". In our opinion, it's at first is an user friendly interface. On the other hand, technical ability to generate arbitrarily complex grammatically correct, but poorly structured and hard to understand sentences is not advantage at all - whether they are expressed in natural or formal language.

For example, in practice it is hardly possible to be sure in correctness of a query in natural language, equivalent in direct or reverse translation to a SQL query with 5–6 attributes, referring to 2–3 database tables. Such considerations lead us to prefer the types of interface based on clearly structured forms of communication, already established in practice. In our case, these are tables containing NL expressions denoting the terms of the subject area.

3 Semantic Search Algorithm

Let's consider now the questions of representing in ontology formalism semantics of RDBs data, critical for the main search algorithm definition. In general, the variety of options here can be reduced to two basic approaches.

The first approach [11] presumes a straightforward conversion of a set of RDB tables into a set of the ontology concepts with slots, corresponding to the table columns, and projecting relations between tables (in the form of migrating keys) into

relations between the corresponding table concepts. An ontology in this case includes also concepts to describe data types. In other words, this approach implies creating an unique ontology for each RDB, which makes process of ontology creation laborious and less universal.

The second approach, which is adopted here, presumes a higher degree of abstraction, in which all the basic theoretical concepts of RDB are described as ontology concepts. Objects (tables, columns, keys and domains) of a particular database in this case are represented as instances of universal concepts of the corresponding type.

Namely, the ontology contains the universal concepts TABLE, COLUMNS, KEY and DOMAIN, corresponding to the main database objects, and the following universal relations:

- TABLE contains the COLUMN;
- TABLE has the primary KEY;
- TABLE has the foreign KEY;
- KEY contains the COLUMN;
- COLUMN values belong to the DOMAIN.

The universal ontology definition includes also two following interpretation functions, playing important role in the semantic search algorithm:

φ1: If TABLE1 has primary KEY1 and TABLE2 has foreign KEY1, then there exists TABLE3, containing all columns of TABLE1 and TABLE2.

φ2: If TABLE1 contains COLUMN1, then there exists TABLE2, containing all columns from TABLE1 except COLUMN1.

The first interpretation function describes the table join operation and the second one describes the relation projection operation, necessary to reduce the set of columns obtained by joining tables to the desired one.

Thus, the task of extracting information from integrated data sources for a given user query can be reduced to finding all the ways to extract specified attributes from RDB tables. In other words, in this case we need to find such sequences of application of φ1 and φ2 functions, which result in the desired set of columns {C} from the ontology O.

Let's especially note here a new point, which significantly distinguishes our approach from the existing works in the field of semantic search [11]. Apart from the existing subject domain ontology and strictly formalized data properties information (such as the RDB schemes), for creating subject domain ontology content we also use information, extracted from informal natural language texts. Namely, those are comments on the names of tables and their attributes. The same information is also used to build the subject domain language thesaurus, which serves as the natural language interface basis. The general approach adopted to solve arising problems of analysis of weakly structured data, using methods of mathematical linguistics, is described in [12].

4 Main System Components

To understand more clear the nature of arising problem of computational complexity, let's have a quick look on the system architecture. The main components of the system are:

- *the RDB ontology*, describing the basic concepts of relational databases in the ontology formalism. For describing ontologies, the OWL language [13], developed by the Semantic Web Activity working group and recommended by the W3C consortium was used [14];
- *the subject domain ontology*. As an initial ontology prototype, the Epicentre data model of Petrotechnical Open Software Corporation (POSC) [15, 16] was used. A general scheme for converting this type of models into the OWL ontology description language using methods of the formal grammars theory has been developed [4, 6];
- *the linguistic thesaurus* of the subject domain language, defining formally the language of the user-system communication;
- *the algorithm of unregulated access to the set of RDBs*, accepting user queries on that natural language dialect.

More detailed description of the general problematic, functionality and architecture of the system can be found in [4, 6].

5 Semantic Search Algorithm Data

The core task of the algorithm is the generation of the text of the SQL-query for a given end user request. The latter has a structure of a table, containing phrases from the linguistic thesaurus in its left column and condition on the values of the corresponding notion in its right column (see Table 1).

Table 1. End user query example.

Professional term	Condition
Oilfield	Novo-Elkhovskoe
Well No.	*
Period	From 01.2010 to 12.2015, monthly
Volume of production, in tons	Sum
Water cut percentage	Average

Thus, the user query language can be considered as a kind of professional dialect of natural language, presented in a form familiar to the end users. Note that such a request, formulated as a sentence in natural language, would be very cumbersome not only for machine analysis, but also for human understanding.

Let's note also, that the user requests refer just to the semantic properties, defined by the subject domain ontology, and contains no references to the logical structure and location of data stored in RDBs.

The most significant and also costly, in terms of computational complexity, part of the algorithm is related to the search of data location, using information on the logical structure of data, described in terms of the RDB ontology.

6 Preprocessing

At this stage, the information on the table joins is extracted from the RDB schemes. As the result, the following graph is constructed from the RDB ontology instance. The vertices of the graph present the "table" RDB ontology concepts, and its arcs are determined by the presence of the common key in the tables; the arcs are oriented in accordance with the "has primary key - has foreign key" relation (see Fig. 1).

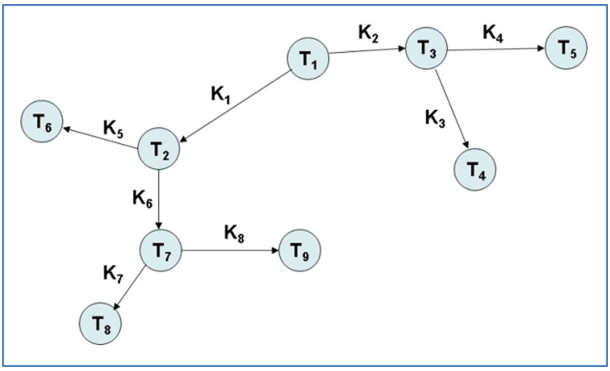

Fig. 1. Graph representation of potential table joins; here T_i are table names and K_i are table joins.

The constructed graph is supplemented by information on possible table joins (up to key migration). This is an algorithmically simple, but resource-intensive procedure, which can be called RDB markup. If the RDB scheme does not change, then this markup is executed once. Otherwise, the above graph must re-build from the new RDB ontology instance.

7 Semantic Analyses of the End User Query

Recall that the subject domain is represented in the system as a semantic network (ontology). During user request analyses we identify in that network all subgraphs, connecting the ontology concepts, corresponding to the phrases of the linguistic thesaurus, used in the user query.

Thus, semantic analyses of the user query is reduced to enumeration of all simple paths, corresponding to some subgraph of the domain ontology in the graph, constructed from the RDB ontology instance (see Fig. 2).

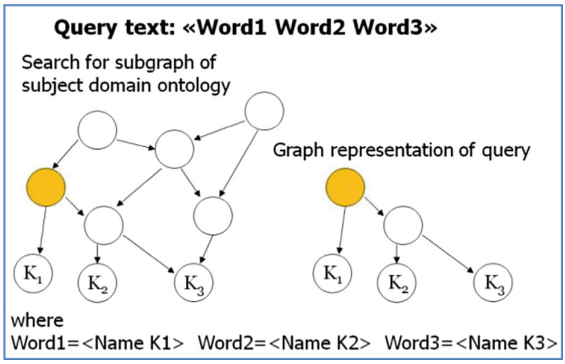

Fig. 2. Process of "understanding" the end user query.

As a result of the semantic analysis, for each column of the user's query several the most relevant locations are found. The corresponding columns can be contained in tables of various RDBs (see Fig. 3). If there are several combinations of the relevant columns, then the number of tables in the constructed join is also taken into account.

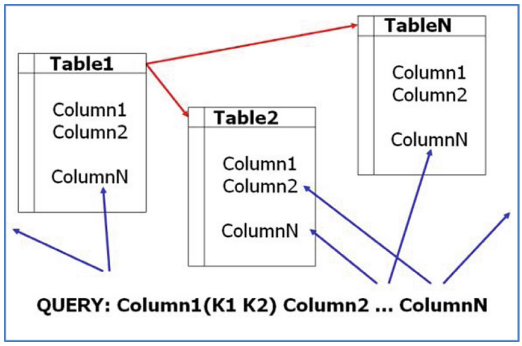

Fig. 3. Search of table joints.

8 On "Fit-for-Purpose" Optimization Approach

The search algorithm described above can be characterized as a "semantically restricted" full search procedure. From a formal point of view, it belongs to the class of graph wandering algorithms, having exponential complexity [17]. Evidently the speed of generation and subsequent execution of the SQL-query essentially depends on the number of columns, contained in it. The main time of the algorithm execution is spent

on finding out the set of RDB tables joins when searching for data location. So the proposed technology in its current implementation works most effectively with the user queries, which refer to a small number of columns; let's note though, that it is quite typical case for corporate RDB queries.

As it follows from said above, the main problem of increasing the system reactivity is connected with the task of reducing the time spent on enumeration of the table joins options.

Let's describe two pragmatic ways to optimization of the alike algorithms, following the "fit-for-purpose" principle, i.e. those aimed on specific situation and purpose of usage. The first, more traditional and universal way presumes improvement of some characteristics of the algorithm (in our case, the speed) for arbitrary input data; it this case we can also take into account expected statistics of usage. The second is based on more specific knowledge of the input data content. Which are in our case the queries made by an oil and gas production specialist.

Optimization of the first type can be implemented as follows. We already store in the system user context, including the history of each user's activity. Practice of the system exploitation shows, that each user usually uses the small number of typical queries related to his/her job duties.

Let's store for each user the sequences of table joins, generated by his/her queries. The analysis of the system functioning shows, that in the overwhelming majority of cases the stored sequence of table joins is sufficient to obtain the desired result in the future. As a rule, the user changes only the query parameters; so the cases, when the search of table joins is required at the last stage of the algorithm execution, are quite rare.

Optimization of the second type takes into account the following specifics of the subject domain. From the beginning of an oilfield development, oil producing companies collect on the regular basis geological survey data to model wells functioning and predict oil reservoir release. As a result RDBs contain a large number of databases and/or tables (up to key migration), that have the same structure and contain the same type of information about the wells exploitation on various calendar periods.

When constructing the graph of table joins during preprocessing, let's store information on presence of date-related (i.e. of year or year-month type) key fields in the considered tables. If later on an user query refers to the temporal characteristics (such as year, year-month, or range of such values), then we will not include into enumeration of table joins the ones which do not match those key values.

9 Conclusions

In the article the rationale for the main semantic search algorithm of data integration systems is given on the example of the actual development of an information system of a large oil mining company. It is noted that there exist theoretical constraints, in the form of the exponential complexity of the algorithm, following from the very statement of the problem. Nevertheless, that leaves room for the successful application of more pragmatic approaches to increase the reactivity of such systems, It can be done either by implementing the well-known effects of "re-pumping complexity" (in our case,

storing the actual query history), or by taking into account the type of information specific to a given subject domain. As the results of experiments show, if without the use of the methods described above, queries containing 6–8 columns of various tables were executed in real time, then with their help it is possible to increase the corresponding number up to 12–15 columns. That is more than enough for operational queries reference.

Acknowledgments. This work was funded by the subsidy allocated to Kazan Federal University for the state assignment in the sphere of scientific activities, grant agreement 1.2368.2018 and subsidy of the Russian fund of fundamental research, grant agreement 18-07-00964.2018.

References

1. Kogalovsky, M.R.: Methods of data integration in information systems. Institut problem rynka RAN, Moscow (2010). http://www.ipr-ras.ru//articles/kogalov10-05.pdf. Accessed 30 Nov 2018
2. Kogalovsky, M.R.: Ontology-based data access systems. Program. Comput. Softw. **38**(4), 167–182 (2012). https://doi.org/10.1134/s0361768812040032. https://link.springer.com/article/10.1134/S0361768812040032. Accessed 30 Nov 2018
3. Kogalovsky, M.R.: Data access systems based on ontologies. Programming, MAIK. "Nauka. Interperiodika" **38**(4), 55–77 (2012). http://www.ipr-ras.ru/articles/kogalov12-03.pdf. Accessed 30 Nov 2018
4. Birialtsev, E., Bukharaev, N., Gusenkov, A.: Intelligent search in big data. J. Phys.: Conf. Ser. **913**, Conf. 1 (2017). http://iopscience.iop.org/article/10.1088/1742-6596/913/1/012010/pdf. Accessed 30 Nov 2018
5. Gusenkov, A.M.: Intelligent search for complex objects in big data arrays. Electron. Lib. **19**(1), 3–39 (2016)
6. Gusenkov, A., Birialtsev, E., Zhibrik, O.: Intelligent search in structured data arrays. LAP LAMBERT Academic Publishing, Deutschland: OmniScriptum Marketing DEU GmbH (2015). ISBN 978-3-659-76919-1
7. SAP Crystal Reports. http://www.crystalreports.com/emea/. Accessed 30 Nov 2018
8. Oracle Fusion Middleware. https://docs.oracle.com/cd/E28280_01/index.htm. Accessed 30 Nov 2018
9. Semantic Search. https://docs.microsoft.com/en-us/previous-versions/sql/sql-server-2012/gg492075(v=sql.110). Accessed 30 Nov 2018
10. Zhigalov, V.A., Sokolova, E.G.: InBASE: technology of building NL-interfaces to databases. Moscow, ROSNII Artificial Intelligence (2001). http://www.dialog-21.ru/digest/2001/articles/zhigalov/. Accessed 30 Nov 2018
11. Zhuchkov, A.V.: New technologies for conceptual networks created in the framework of the ICST "New generation vaccines and diagnostic systems of the future". Electron. Lib. **6** (2003). https://elbib.ru/ru/article/244. Accessed 30 Nov 2018
12. Birialtsev, E.V., Gusenkov, A.M., Mironov, S.V.: One approach to implementing unregulated access to relational databases. In: Trudy Kazanskoj shkoly po komp'yuternoj i kognitivnoj lingvistike TEL-2008, pp. 10–23. Kazanskij gosudarstvennyj universitet, Kazan (2009)
13. OWL Web Ontology Language. https://www.w3.org/TR/2004/REC-owl-features-20040210/. Accessed 30 Nov 2018

14. World Wide Web Consortium (W3C). https://www.w3.org/. Accessed 30 Nov 2018
15. Epicentre v3.0. http://www.energistics.org/energistics-standards-directory/epicentre-archive. Accessed 30 Nov 2018
16. Petrotechnical open standards consortium (Energistics). http://www.energistics.org. Accessed 30 Nov 2018
17. Anderson, J.A.: Discrete Mathematics with Combinatorics, 2nd edn., p. 784. Prentice Hall (2003). ISBN 0130457914