Volume 5, Issue 4, April 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Retreiving Unstructured Data from RDBMS Using Natural Language Queries

Cheshta, Tarun Bagga
Computer Science & Engineering, HEC, Jagadhri
Haryana, India

Abstract— (Natural language processing is a field of computer science concerned with the interactions between computers and human (natural) languages. The main purpose of Natural Language Query Processing is for an English sentence to be interpreted by the computer and appropriate action taken. Here, we use the artifacts of NLP to convert the user query in natural language into a SQL query based on the query entered. We will define an interface (IRDBNLQ) where the user enters his query and then the system parses the query and converts it into a SQL query and executes it to display the intended result. We design a new language (XML based) i.e QML whose DTD specifies the tags permitted in the language. Then the query parser creates a XML for the user query and finally, the XML is converted into SQL query. The concept will be implemented in java language, which allows easy and quick XML parsing. The conversion into QML makes the query exportable into many other formats and can be used for different purposes.

Keywords— XML, DTD, RDBMS, SQL, QML.

I. INTRODUCTION

Natural language processing is a field of computer science concerned with the interactions between computers and human (natural) languages. It is becoming one of the most active areas in the interaction between human and computer. These include spoken language systems that integrate speech and natural language. It is an interdisciplinary research area at the border between linguistics and artificial intelligence, aiming at developing computer programs capable of human-like activities like understanding or producing texts or speech in a natural language, such as English or conversion of natural language in text or speech form to languages like SQL. The most important applications of natural language processing include information retrieval and information organization, machine translation. The goal of NLP is to enable communication between people and computers without resorting to memorization of complex commands and procedures. A huge amount of labour is required if we wish to obtain only the required information from the entire repository of information system. Natural language processing is a process by which the user query (entered in English language) in natural language will be converted to a SQL query based on the query entered. Asking questions to databases in natural language is a very convenient and easy method of data access, especially for casual users who do not understand complicated database query languages such as SQL. Any ordinary person is not expected to know the SQL language, and hence this system would help him in generating the same, so that information retrieval is easier for the database, as database understand the SQL language only.

Fig 1 Demonstration of IRDBNLQ process

Assume that a business man wants the top 10 customer details from Haryana who have more than Rs.100000 debited to their name.

So the SQL query should be

RDBMS> SELECT * FROM customers
WHERE state='Haryana' and bal > 100000 SORT BY bal ASC LIMIT 10;

This is the conversion we expect our tool to perform wherein the user inputs natural language query in written or oral form and gets the desired results from the database via a SQL query.

II. LITERATURE SURVEY

Remarkable work has been already done in the field of natural language processing. Here we review some of the previous studies and researches related to our work of SQL query generation from natural language query.

Androutsopoulous et.al in 1995 [1] has discussed some of the linguistic problems such conversion systems have to confront. He then discussed the Nlidb architectures, portability issues, restricted natural language input systems (including menu-based Nlidbs), and Nlidbs with reasoning capabilities. Some less explored areas of Nlidb research were then discussed, namely database updates, meta-knowledge questions, temporal questions, and multi-modal Nlidbs.

Ana Maria Popesqu et. al [2] in 2003 introduced a theoretical framework for reliable NLIs which was then the foundation for fully implemented precise NLI.

Esther Kaufmann et.al in 2010 [3] introduced four interfaces each allowing a different query language and present a usability study benchmarking these interfaces. The results of the study reveal a clear preference for full natural language query sentences with a limited set of sentence beginnings over keywords or formal query languages.

Georgia Koutrica et.al in 2010 [4], represented various forms of structured queries as directed graphs and then annotated the graph edges with template labels using an extensible template mechanism. He presented different graph traversal strategies for efficiently exploring these graphs and composing textual query descriptions. Finally, he presented experimental results for the efficiency and effectiveness of the proposed methods.

Lukas Blunschi et.al in 2012 [5], described the design, implementation, and experience of the SODA system (Search over Data Warehouse). SODA bridges the gap between the business needs of analysts and the technical complexity of current data warehouses. SODA enables a Google-like search experience for data warehouses by taking keyword queries of business users and automatically generating executable SQL.

In 2013, Yingzhong Xu et.al, in his research, [6] developed a tool named QMapper to map SQL queries into SQL queries by utilizing query rewriting rules and cost-based MapReduce flow evaluation on the basis of column statistics. Evaluation demonstrated that while assuring the correctness, QMapper improved the performance up to 42% in terms of execution time by extending the Rewriter to support more rules and enhancing the estimator to involve the parallelism of multiple SQL blocks, data compression and the difference of cost effect between Map and Reduce to provide a more fine-grained cost model.

Rakesh Kumar et.al, in 2014, in his research,[6] performed a comparison between SQL and SQL on basis of their architecture, advantages, disadvantages and other features. The author concluded that the primary reason for moving data between SQL stores as well as Hadoop was usually to take advantage of the massive storage and processing capabilities to process quantities of data larger than one could hope to cope with in SQL alone. Prasun kanti Ghosh et.al, in 2014, [7] created a natural language processing tool that integrated speech and natural language. The author worked to enable communication between people and computers without resorting to memorization of complex commands and procedures

III. PROBLEM FORMULATED

Querying data in relational databases is often challenging. SQL is the standard query language for relational databases. While expressive and powerful, SQL is too difficult for users without technical training. Even for users with expertise in programming languages, it can be challenging because it requires that users know the exact schema of the database, the roles of various entities in a query, and the precise join paths to be followed. As the database user base is shifting towards non-experts, designing user-friendly query interfaces will be a more important goal in database community. So, to fulfill the needs of these non-technical people, we need a way to convert their natural queries into SQL queries and fetch desired results.

SCOPE OF PROPOSED SYSTEM:-

The scope of the proposed system is as follows:

- To work with RDBMS one should know the syntax of the commands of SQL.
- The interface language is chosen to be English for accommodating wider users.
- Input from the user is taken in the form of questions (like what, who, where).
- All the values in the input natural language statement have to be in double quotes which yield to identify the values from the user input statement.
- A limited data dictionary is used where all possible words related to a particular system will be included. The data dictionary of the system must be regularly updated with words that are specific to the particular system.
- Split the question string in to tokens and give order number to each token identified.
- To remove excessive words from the user input statement. Escape words have been considered which must be regularly updated with words that are specific to the particular system.
- To construct an RDBMSQL query using tokens, an algorithm has to be developed.
- Ambiguity among the words will be taken care of while processing the natural language.
- all text labels in each figure are legible.

IV. OBJECTIVE

The main objective of this work is to bring forward a novel scheme to convert natural language queries into SQL queries. To resolve this issue, we studied the above listed literatures and finally we aim to create a method which can parse a natural language statement, convert it to tokens and then finally generate the resultant SQL query. The problem arrives when the table names are clear from the statement or the query intended requires a join.

V. PROPOSED METHODOLOGY

Converting a natural language statement into SQL query requires the proper table names, fields, constraints or mixed expressions to be identified. To do this, we can break the query into tokens and then by understanding of prepositions, we can make out which tokens correspond to which components in SQL query. So there must be a mapping method to map tokens with RDBMS database entities. In this method we define a new language called QML based on XML. The DTD (Document type definition) of these XML files defines the tags permitted and their subtags which correspond to entities in the database queries. Here we also define a mapping function which maps words retrieved from NLQs to the entities and their attributes in the database. The following steps detail the methodology we are going to adopt for this work.

- Accept the natural language query from the user.
- Split the query to generate a list of tokens.
- Remove the tokens from the list which don't have relevance in database context.
- Perform stemming of each word to make mapping easier.
- Scan the list from beginning to end and call mapping function for each word which maps the token to a RDBMS database object or attribute or condition or a clause.
- Keeping in mind the rules defined in DTD, create the XML for the present query.
- Last, use the conversion function defined to convert the XML into SQL query.
- Execute the query to get results and display results.

VI. CONCLUSIONS

Here, we define a novel technique to convert the natural language queries into SQL queries. Conversion to XML is always a better method than previous techniques where we convert the queries into graphs and then graphs into SQL queries [4]. This is because XML parsing is far more easier and efficient than graph traversal. So the introduction of QML will certainly enhance the performance. In future we can define methods to export QML into other commonly used formats to increase the acceptability of the language.

REFERENCES

[1] Androutsopoulous, Ritchie and Thanisch in "Natural language interfaces to databases-An introduction" Arxiv,Journal of natural language processing, Cambridge university press, P. No. 709 Vol. 2, 1995.

[2] Ana Maria, Oren Etzioni and Henry Kautz in "Towards a theory of natural language interfaces to databases", Springer ACM Pg. 586-594, 2003.

[3] Georgia Koutrica, Alkis Simitsis and Yannis E. Loannidis in "Explaining structured queries in natural language", IEEE, 2010.

[4] Esther Kauffmann and Abraham Bernstein in "Evaluating theUsability ofNatural LanguageQuery Languages and Interfaces to Semantic Web Knowledge Bases", Journal of web semantics, 2010.

[5] Lukas Blunschi, Claudio Jossen and Donald Kossman in "SODA: generating SQL for business users", International conference on very large databases, Vol. 5 No. 10, 2012.

[6] Rakesh Kumar, Neha Gupta, Shilpi Charu, Somya Bansal and Kusum Yadav, "Comparison of SQL with SQL", International Journal for Research in Technological Studies| Vol. 1, Issue 9, August 2014. Pp. 28-30.

[7] Prasun Kanti Ghosh, Saparja Dey and Subharta Sengupta in "Automatic SQL formation from Natural Language Query" International conference on micro-electronics, circuits and systems, 2014.

