Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 92 (2016) 487 – 492

Procedia Computer Science

2nd International Conference on Intelligent Computing, Communication & Convergence

(ICCC-2016)

Srikanta Patnaik, Editor in Chief

Conference Organized by Interscience Institute of Management and Technology

Bhubaneswar, Odisha, India

Ontology Based Natural Language Interface for Relational Databases

B.Sujatha*[a], Dr.S.Viswanadha Raju[b]

a. *Assistant Professor, Department of CSE, Osmania University, Hyderabad, India*

b. *Professor, Dept. Of CSE, JNTU college of Engineering, Jagityal, Karimnagar, India*

Abstract

Developing Natural Language Query Interface to Relational Databases has gained much interest in research community since forty years. This can be termed as structured free query interface as it allows the users to retrieve the data from the database without knowing the underlying schema. Structured free query interface should address majorly two problems. Querying the system with Natural Language Interfaces (NLIs) is comfortable for the naive users but it is difficult for the machine to understand. The other problem is that the users can query the system with different expressions to retrieve the same information. The different words used in the query can have same meaning and also the same word can have multiple meanings. Hence it is the responsibility of the NLI to understand the exact meaning of the word in the particular context. In this paper, a generic NLI Database system has proposed which contains various phases. The exact meaning of the word used in the query in particular context is obtained using ontology constructed for customer database. The proposed system is evaluated using customer database with precision, recall and f1-measure.

Peer-review under responsibility of the Organizing Committee of ICCC 2016

Keywords: Natural Language queries, Ontology, Relational Databases, First Order Logic, Structured Language Query, Backus Naur Form, N-grams.

*Corresponding Author. Tel: 08977207889

E-Mail Address: drpvijayapalreddy@gmail.com

1. Introduction:

Database systems are used since 1970s for the storing various kinds of data for different purposes such as commercial and personal needs. Though there are many types of architectures for database design like object oriented, object based, file based, hierarchical based and network based, the predominant designing of databases follow relational database architecture to store the data by using various types of storage devices. In relational databases, the data is stored using tables. The table contains set of rows and columns. Each column represent and attribute and each represents the instance of the data for a set of attributes. The data can be manipulated using various operators with fixed set of keywords by following a set syntax rules. By learning this structured query language one can extract the required data from the whole set of data, can also perform various operations such as update, manipulate and deletion of the data.

Peer-review under responsibility of the Organizing Committee of ICCC 2016

doi:10.1016/j.procs.2016.07.372

The Relational database management systems are more popular based on the characteristics like its robustness and flexibility, high performance, scalability, data security and protection and flexible data maintenance. Above all these advantages, it allows to index, perform aggregation, filtering and sorting can be done on the data using structured query language.

There are some disadvantages with relational databases. To perform operations on the data which is stored on databases, it is required to learn the structured query language. Hence , the naive user who knows only the natural language can not directly access the required information from the databases. To come out from these limitations, it is required to design a tool which can understand the requirements of the naive user through natural language query, convert the natural language query into an equivalent structured language query. Then the obtained structural query is used to access the required information from the databases. This kind of tool ins termed as Natural Language Interface to Databases or NLIDB system. Thus, the NLIDB system take the input as natural language query and converts it into a structures language query and returns the desired information to the naive user.

The designing of a NLIDB system for various languages and for different underlying databases is attempted by various researchers since five decades. But, designing of an most suitable NLIDB systems with high accuracy, precision and recall is still an open research problem which need to be addressed. The various earlier developed NLIDB systems focused on particular databases. There is need of designing a generic NLIDB system which can address the robustness and scalability of the applications. It is required to attempt the problem of portability to customize a NLIDB system to a other language and to other set of datasets designed for various domains. The efficiency of conventional NLIDB systems depend mostly on domain experts capabilities and linguistic features of the natural language.

In this paper, it is focused on designing a NLIDB system to overcome the various issues such as portability to different languages and to access the required information independent of the underlying database. It also required maintains the scalability and robustness of the system. The word sense disambiguation is achieved using N-grams and ontology which is constructed on the customer database. The remaining sections explains about the related work, proposed model, word sense disambiguious using N-gram and ontology, experimental evaluation and conclusions and possible future extensions.

2. Related Work:

There are many designing models are proposed in the literatures in the field of NLIDB such as pattern matching systems, syntax based systems, semantic based grammar systems and intermediate representation of languages system. The pattern matching systems takes input as a set of rules and sample set of patterns. Based on the inputted word of sentence with natural language, it will be compared with the predefined patterns [1]. If there is a match between the input and predefined pattern then an action will be generated and these generated actions will be stored in the database. The response given to the user is based on the action generated. This kind of systems are limited to specific databases. The accuracy of the system is depend on the complexity of the patterns used to train and based on the set of rules used to train the system [2]. The NLIDB system SANVY is a good example for pattern- matching systems [3].

The syntax based systems takes the user query as input and parse the given input syntactically. The parse tree generated for the input query is overlapped with the one structured query of the database expressed using structured query language. LUNAR is a best example for syntax based NLIDB systems [4]. In these systems, the grammar rules are derived to match the various user questions with syntactic structures [5]. This system is used to answers the questions on rocks which were collected from the moon. With the corrections in the dictionary errors, the performance of the system has increased [8].

In the semantic grammar system, the parse is simplified by eliminating unimportant nodes or by combining two or more nodes into one node. The complexity of structured query can be reduced in semantic grammar system. Semantic grammar systems are more simpler when compared with syntax based systems. But these systems need to be trained with a prior knowledge of the various elements of a domain. PLANES and LADDER are the good examples for Semantic grammars systems [6,7].

In many NLIDB systems, the natural language query is transformed into an intermediate logical query. The logical query is represented using a meaningful representative language such as first logic language or Boyce Codd normal form. This kind of representative languages, represents the meaning of the users queries in high order level of concepts. These concepts are independent from the structure of the database. This representative query is then transformed into an expression in the structured query language which can extract the relevant data from the databases.

In the intermediate representation of natural language systems, the natural language query is inputted to the system. This query is processed for syntax rules using a parser. Based on the set of syntax rules of a natural language, it generates a parse tree. By using the semantic rules of semantic interpreter module, the generated parse tree is translated into an intermediate logic query. In the semantics rule, left hand side of the syntax rule contains the logic expression of the constituent where as right-hand side of the syntax rule is a function of the logic expressions of the constituents. The logic expressions represents the words which are corresponds to lexicon. To get the required information from the database, the logic query is to be transformed into a structured query which is supported by the underlying Database Management System. MASQUE/SQL is an example of intermediate