Natural Language Interfaces to Database (NLIDB)

Shripad Pramod Rane, Akanksha Kulkamil
SOE (School Of Engineering)
AJEENKYA D.Y.PATIL UNIVERSITY, PUNE, INDIA

Abstract:- Now a days, Information is playing an important role in our lives. One of the major sources of information is databases. Databases and database technology are having major impact on the growing use of computers. Almost all IT applications are storing and retrieving information from databases. Retrieving information database requires knowledge of database languages like SQL. The Structured Query Language (SQL) norms are been pursued in almost all languages for relational database systems. However, not everybody is able to write SQL queries as they may not be aware of the structure of the database. The idea of using Natural Language instead of SQL has prompted the development of new type of processing called Natural language Interface to Database.

Keywords:- *Databases, Database Management System (DBMS), Structured Query Language (SQL), Natural Language Interface for Databases (NLIDB), Flexible Querying.*

I. INTRODUCTION

A natural language interface to a database (NLIDB) is a system that allows the user to access information stored in a database by typing requests expressed in some natural language (e.g. English), or a subset of natural language. Database access has been increasing rapidly these years, as every business element use database for data storing. Databases are gaining prime importance in a huge variety of application areas employing private and public information systems. A database model used widely is the relational database model, which uses Structured Query Language (SQL) as a means of getting the data stored inside. The Structured Query Language (SQL) norms are been pursued in almost all languages for relational database systems. To ease the use of relational database, interfaces are made so people don't have to understand SQL to access the database. One of the interfaces is Natural Language Interfaces to Database, or NLIDB. In the early development of NLIDB, the system relied heavily on linguistic and domain experts for its creation or customization, hence making the portability either expensive or sometimes impossible [1]. A recent development was introduced by Wibisono [2]. who made a NLIDB system with ontology as semantics processing and Stanford Dependency Parser as the natural language analyzer. Wibisono's NLIDB system was only able to process declarative-type query, e.g. "Get employees whose last name is Smith" Therefore the idea of using natural language instead of SQL has prompted the development of new type of processing method called Natural Language Interface to Database systems (NLIDB). NLIDB is a step towards the development of intelligent database systems (IDBS) to enhance the users in performing flexible querying in databases.

II. NLIDB (NATURAL LANGUAGE INTERFACE TO DATABASES)

Persons with no knowledge of database language may find it difficult to access database. In recent time there is a rising demand for non-expert users to query relational database in a more natural language encompassing linguistic variables and terms. Therefore, the idea of using natural language instead of SQL triggered the development of new type of processing method Natural Language Interface to Database. Although the earliest research has started since the late sixties [3], NLIDB remains as an open research problem. A complete NLIDB system will benefit us in many ways. Anyone can gather information from the database by using such systems. Additionally, it may change our perception about the information in a database. Traditionally, people are used to working with a form; their expectations depend heavily on the capabilities of the form. therefore, will maximize the use of a database.

eural models have become the standard approach for NLIDB solutions. The typical neural architecture consists of an encoder and a decoder component, in a Seq2Seq approach. In this section, we explore the most significant design considerations for the two components, as presented in recent literature. eural models have become the standard approach for NLIDB solutions. The typical neural architecture consists of an encoder and a decoder component, in a Seq2Seq approach. In this section, we explore the most significant design considerations for the two components, as presented in recent literature. eural models have become the standard approach for NLIDB solutions. The typical neural architecture consists of an encoder and a decoder component, in a Seq2Seq approach. In this section, we explore the most significant design considerations for the two components, as presented in recent literature eural models have become the standard approach for NLIDB solutions. The typical neural architecture consists of an encoder and a decoder component, in a Seq2Seq approach. In this section, we explore the most significant design considerations for the two components, as presented in recent literature eural models have become the standard approach for NLIDB solutions. The typical neural architecture consists of an encoder and a decoder component, in a Seq2Seq approach. In this section, we explore the most significant design considerations for the two components, as presented in recent literature

III. METHODOLOGY

Neural models have become the standard approach for NLIDB solutions. The typical neural architecture consists of an encoder and a decoder component, in a Sequence to Sequence approach. In this section, we explore the most significant design considerations for the two components, as presented in recent literature.

eural models have become the standard approach for NLIDB solutions. The typical neural architecture consists of an encoder and a decoder component, in a Seq2Seq approach. In this section, we explore the most significant design considerations for the two components, as presented in recent literature

IV. SUB COMPONENTS OF NLIDB

Scientists have divided the problem of natural language access to a database into two sub-components:

- Linguistic component
- Database component

A. Linguistic component

Linguistics is the scientific study of human language. It is called a scientific study because it entails a comprehensive, systematic, objective, and precise analysis of all aspects of language, particularly its nature and structure. Linguistics is concerned with both the cognitive and social aspects of language. It is considered a scientific field as well as an academic discipline; it has been classified as a social science, natural science, cognitive science, or part of the humanities.

B. Database Component

It performs traditional Database Management functions. A lexicon is a table that is used to map the words of the natural input onto the formal objects (relation names, attribute names, etc.) of the database. Both parser and semantic interpreter make use of the lexicon. Computer scientists may classify database management systems according to the database models that they support. Relational databases became dominant in the 1980s. These model data as rows and columns in a series of tables, and the vast majority use SQL for writing and querying data. In the 2000s, non-relational databases became popular, collectively referred to as NoSQL, because they use different query languages.

C. Various Approaches Used for Development of NLIDB Systems

Natural language is the topic of interest from computational viewpoint due to the that language possesses. Several researchers applied different techniques to deal with language.

- Symbolic Approach (Rule Based Approach).
- Empirical Approach (Corpus Based Approach).
- Connectionist Approach (Using Neural Network)

V. ADVANTAGES AND DISADVANTAGES OF NLIDB[4]

A. Advantages: -

- No Artificial Language
- Simple, easy to use
- Better for Some Questions
- Fault tolerance
- Easy to Use for Multiple Database Tables

B. Disadvantages: -

- Linguistic coverage is not obvious.
- Linguistic vs. conceptual failures.
- False expectations

C. Architecture of NLIDB systems

This section describes architectures adopted in existing systems.[5]

- Pattern Matching systems.
- Syntax-Based Systems.
- Semantic Grammar Systems.
- Intermediate Representation Languages.

D. History of NLIDB

Over the past 60 years, many attempts have been made to create intelligent Natural Language Interfaces for querying databases. The very first attempts at NLP database interface are just as old as any other NLP research. Asking database to NLIDB is very easy to understand the method of data access. Those who do not understand SQL. They will learn easily.

There are some five example of NATURAL LANGUAGE INTERFACE OF DATABASE.[6]

- LUNAR system (1971).
- LADDER
- CHAT-80. (MID-1980).
- DATALOG.
- PLANES.

VI. CONCLUSION

Research is done from the last few decades on Natural Language Interfaces. Though several NLIDB systems have also been developed so far for commercial use but the use of NLIDB systems is not wide-spread and it is not a standard option for interfacing to a database. NLIDB is a very active field in automatic language processing. Its purpose is to accept requests expressed in natural languages often used by non-technical users and to generate responses. It is a type of human-machine interface. Most of the tools developed are very efficient and have obtained encouraging results, but unfortunately, the majority of them have been developed so far for a specific use. Research in NLIDB is still in its infancy and needs to be continued. The use of NLIDB systems is not widespread and is not the optimal option for querying databases. This is mainly due to a large number of deficiencies in the NLIDB systems and the lack of a generic model that meets user expectations.

REFERENCES

[1.] Gupta, Abhijeet, et. al. (2012). A Novel Approach Towards Building a Portable NLIDB System Using The Computational Paninian Grammar Framework. 2012 International Conference on Asian Language Processing

[2.] Wibisono, Okiriza. (2013). Pengambilan Data dari Basis Data RelasionalDengan Query Bahasa Alami Menggunakan Stanford Dependency Parser. Bandung: Tugas Akhir, Institut Teknologi Bandung.

[3.] Androutsopoulos, G.D. Ritchie, and P. Thanisch, Natural Language Interfaces to Databases - An Introduction, Journal of Natural Language Engineering 1 Part 1 (1995), 29–81.

[4.] Kaur, and P. Bhatiya, "Punjabi Language Interface to Database", M.tech thesis, Department of CSED, Thapar University, 2010.

[5.] https://www.nexthink.com/blog/natural-language-interfaces-databases-nlidb/

[6.] https://www.researchgate.net/publication/348747338_The_history_and_recent_advances_of_Natural_Language_Interfaces_for_Databases_Querying