National Conference on Recent Research in Engineering and Technology (NCRRET-2015)
International Journal of Advance Engineering and Research Development (IJAERD)
e-ISSN: 2348 - 4470 , print-ISSN:2348-6406

A Survey: Natural Language Interface to Databases

Jaina Patel*[1], Jay Dave#[2]
* *Assistant Professor* in CSE Department, DJMIT, Gujarat, India. j4jaina@gmail.com
# *Assistant Professor* in CSE Department, DJMIT, Gujarat, India. jaydavejob@gmail.com

Abstract-Information plays an important role in our everyday life and databases are widely used for storing and retrieving information. Database technology is having major impact in the world of computing. To access the information from database one need to have knowledge of database query language such as SQL. Because the naïve user may not be aware of the syntax of SQL and structure of database, s/he may not be able to write the SQL queries. Non-technical users may query relational databases in their natural language (i.e. English) instead of using SQL. This idea of using a Natural Language instead of SQL has lead to an approach of building Natural Language Interface to Relational Database. This paper is an introduction to the natural language interface to databases (NLIDB).

***Index Terms*-Natural language interface, database, computer-human interface, natural language processing**

I. INTRODUCTION

Database systems are designed to manage large collection of information. To access this information, user should have the knowledge of Structured Query Language (SQL). Only those users who have the knowledge of these languages can access the data or information [2]. Normally end-users have no knowledge of SQL so a graphical user interface is required to access information. By using this interface the end-user can query the system in native language like English. This gives the idea of Natural Language Interface to Database (NLIDB). A natural language interface to a database (NLIDB) is a system where the users access information stored in a database by typing requests in some natural language (e.g. English) [1]. NLIDB system is proposed as a solution to the problem for accessing information in a simple way allowing ideally any type of users, mainly inexperienced ones, to retrieve information from a database (DB) using natural language (NL) [3]. It is a type of computer human interface. This is the user-friendly interface through which users can interact with the database [4]. A complete NLIDB system will provide many benefits. Anyone can retrieve information from the database by using such system. Traditionally, people are very familiar working with a form. In devices like PDA and cell phone, the display screen is not much wide as a computer or a laptop. For example filling a form is a very uninteresting work; users have to scroll through the screen, to select choices, radio buttons, fill text fields etc. Instead, with NLIDB, they only have to type the question similar to the English question.

II. ADVANTAGES AND DISADVANTAGES OF NLIDB

Advantages of NLIDB

- No requirement of Artificial Language.
- No need of Training.
- Simple and easy to use.
- Better for some question.
- Tolerances to minor grammatical errors.[2]

Disadvantages of NLIDB

- Deals with limited set of natural language.
- Linguistics coverage is not obvious.
- Linguistics vs. conceptual failure.
- Users assume intelligence.
- Tedious configuration.[2]

III. VARIOUS APPROACHES FOR NLIDB SYSTEMS

There are various approaches for developing NLIDB systems:

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

Symbolic Approach (Rule Based Approach): The Natural Language process is strongly a symbolic activity. Here language is analyzed and rule based reasoning captures the meaning of language based on the rules. [5]

The knowledge related to the language is explicitly encoded in rules or other representation forms. Language is analyzed at various levels to obtain information. Certain rules are applied on this obtained information to achieve linguistic functionality. In symbolic approach rules are created for every level of linguistic analysis. Based on these rules, the meaning of the language is analyzed. [2]

Empirical Approach (Corpus Based Approach): Empirical approach follows statistical analysis and other data driven analysis of raw data. Collection of machine readable data which are primarily used as a source of information about language is known as corpus. There are various techniques for analyzing the texts data. Statistical probabilities estimated from a training corpus and based on it the syntactic analysis can be achieved. Various statistical techniques such as n-gram models, hidden Markov models (HMMs) and probabilistic context free grammars (PCFGs) are employed in major empirical NLP methods. [2]

Connectionist Approach (Using Neural Network): As human language capabilities are based on neural network in the brain, connectionist network (artificial neural network) provides an essential starting point for modeling language processing. Sub-symbolic neural network approach has lot of ability to model the cognitive foundations of language processing. This approach is based on distributed representations corresponding to statistical regularities in language [2].

IV. EXISTING ARCHITECTURAL FRAMEWORKS

This section describes architectures adopted in existing systems.

- Pattern-matching systems
- Syntax-based systems
- Semantic grammar systems

Pattern-matching Systems

To illustrate a simplistic pattern-matching approach, consider a database table holding information about countries[1]:

Table 1 Country Information

COUNTRIES_TABLE		
COUNTRY	CAPITAL	LANGUAGE
India	Delhi	Hindi
France	Paris	French
. . .	. . .	. . .

A primitive pattern-matching system could use rules like: [1]

Pattern: ... “capital” ... <country>

Action: Report CAPITAL of row where COUNTRY=<country>

Pattern: ... “capital” ... “country”

Action: Report CAPITAL and COUNTRY of each row

The first rule says that if a user’s request contains the word “capital” followed by a country name (i.e. a name appearing in the Country column), then the system should locate the row which contains the country name, and print the correspond ding capital.

According to the second rule, any user request containing the word “capital” followed by the word “country” should be handled by printing the capital of each country, as listed in the database table. SAVVY and ELIZA are the systems that are based on pattern-matching.

Advantage:

- Easy to implement.
- Easy to add or subtract features by just adding more patterns.[23]

Disadvantage:

- It is too shallow, only matches for limited patterns. [23]

Syntax-based systems

In syntax-based systems the user’s question is parsed (i.e. analyzed syntactically), and the resulting parse tree is directly mapped to an expression in some database query language. [1] LUNAR is syntax-based system. [3] Syntax-based systems use a grammar that describes the possible syntactic structures of the user’s question. The following example shows a simplistic grammar in a Lunar-like system.

S→NP VP

NP→Det N

Det→“what”| “which”

N→“rock”| “specimen”| “magnesium”| “radiation”| “light”

VP→V N

V→“contains”|“emits”

The grammar above says that a sentence (S) consists of a noun phrase (NP) followed by a verb phrase (VP), that a noun phrase consists of a determiner (Det) followed by a noun (N), that a determiner may be “what” or “which” , etc. Using this grammar, a NLIDB could figure out that the syntactic structure of the question “which rock contains magnesium” is as shown in the parse tree of Fig. 1.

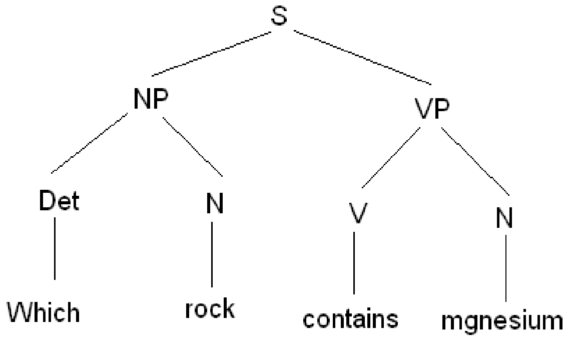

Fig. 1: Parse tree in a syntax-based system

The NLIDB could then map the parse tree of Fig. 1 to the following database query (X is a variable):

(for_every X (is_rock X)

(contains X magnesium) ;

(printout X))

which would then be evaluated by the underlying database system by using the mapping rules.[1]

Advantage:

- It provides detailed information about the structure of sentence.[23]

Disadvantage:

- Not clear which node should be mapped.
- It is difficult to directly map a tree into general database query.[23]

Semantic Grammar Systems

In semantic grammar systems, the question-answering is still done by parsing the input and mapping the parse tree to a database query. The difference, in this case, is that the grammar’s categories (i.e. the non-leaf nodes that will appear in the parse tree) do not necessarily correspond to syntactic concepts. Following is a possible semantic grammar:

S→Specimen question|Spacecraft question

Specimen_question→Specimen_spec Emits_info|Specimen_spec Contains_info

Specimen→“which rock”|“which specimen”

Emits_info→“emits” Radiation

Radiation→“radiation”|“light”

Contains_info→“contains” Substance

Substance→“magnesium”|“calcium”

Spacecraft_question →Spacecraft Depart_info|Spacecraft Arrive_info

Spacecraft→“which vessel”|“which spacecraft”

Depart_info→“was launched on” Date|“departed on” Date

Arrive_info→“returns on” Date|“arrives on” Date

Following Fig. 2 is the possible parse tree:

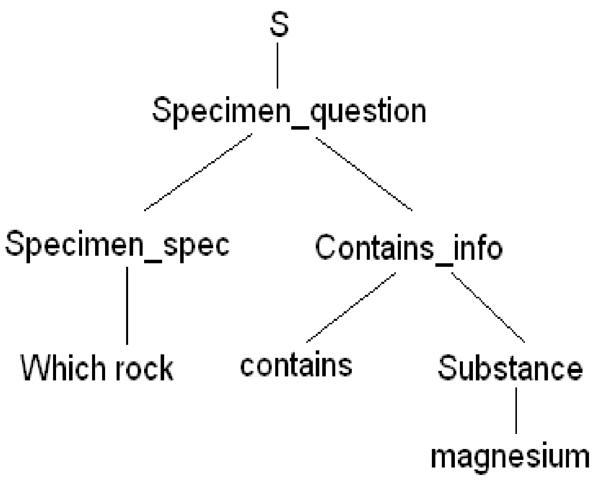

Fig. 2: Parse tree in a semantic grammar

Advantage:

- Less ambiguity[23]

Disadvantage:

- It requires prior –Knowledge of the elements in the domain making it difficult to port to other domain.[23]
- Its specific structure could hardly be used for other application.[23]

V. EARLIER PROGRESS OF NLIDB

Research in Natural Language Interface for Relational Databases has started in 20th century. The first Natural Language Interface for Relational Databases appeared in the 1970s [6], the NLIDB system was called LUNAR. After the first NLIDB, many were developed that supposed to improve the apparent flaws of LUNAR [7]. Below Table 2 highlights the development of some NL interfaces:

Table 2 Earlier NLIDB Systems

System name & Year	Domain	Language	Approach	Technique
LUNAR (1973) [8]	Rock samples from moon	English-SQL-English	Connectionist (neural network) [9]	Syntax-based system
LADDER (1978) [10]	US-Navy ships	English-SQL-English	Empirical (Corpus based)	Semantic grammar system [11]
CHAT-80 (1980) [12]	General	English-Prolog-English	Dialogue based	Semantic grammar system
JANUS (1989)	General	English	Menu based	ER-based intermediate representation
PRECISE (2004) [13]	Air Travel Information System & GEOQUERY	English-SQL-English	Lexical analysis and semantic constrains	Keyword matching and semantically tractable sentences
WASP (2005) [14]	GEOQUERY	English-Prolog-English	Corpus based	Semantic parsing and statistical machine translation
NALIX (2006) [15]	XML database	English-XQuery-English	Keyword search in XML database	Syntax-based reverse-engineering [16]
GINLIDB (2009) [17]	General	English-SQL-English	Lexical analysis and Syntactic analysis	Augmented Transition Network and Context-Free Grammar [18] [19]
Punjabi Language Interface to Database (2010) [20]	Agriculture	Punjabi-SQL-Punjabi	Shallow parsing	Mapping Punjabi language words to English words
Hindi Language Interface to Database (2011) [21]	Employee	Hindi-SQL-Hindi	Shallow parsing	Mapping Hindi root words with corresponding English words
Intelligent Query Converter (2013) [22]	General	English-SQL-English	Semantic analysis	Semantic matching

NaLIR (2014) [24]	General	English-SQL	Dependency Parser	Parse tree and mapping

VI. CONCLUSION

This paper has attempted to serve two purposes: to introduce the reader to the field of NLIDBs by outlining the facilities and methods of typical implemented systems. The goal of surveying the field can be achieved only incompletely at any given moment. Research is done from the last few decades on Natural Language Interfaces. With the advancement in hardware processing power, many NLIDBs mentioned in historical background got promising results. 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. This lack of acceptance is mainly due to the large number of deficiencies in the NLIDB system in order to understand a natural language.

REFERENCES

[1] I. Androutsopoulos, G.D. Ritchie, and P. Thanisch, “Natural language interfaces to databases-An Introduction”, in arXiv preprint cmp-lg/9503016, 1995

[2] N. Nihalani, S. Silakari, and M. Motwani, “Natural language Interface for Database: A Brief review” in International Journal of Computer Science Issues (IJCSI), 8(2), 2011

[3] N. Nihalani, M. Motwani, and S. Silakari, “Natural Language Interface to Database using Semantic Matching”, in International Journal of Computer Applications, 31, 2011

[4] M. Owda, Z. Bandar, and K. Crockett, “Conversation-based natural language interface to relational databases”, in Web Intelligence and Intelligent Agent Technology Workshops, 2007 IEEE/WIC/ACM International Conferences on (pp. 363-367), November, 2007

[5] R. Miikkulainen, “Subsymbolic natural language processing: An integrated model of scripts, lexicon, and memory”, MIT press, 1993

[6] A.M. Popescu, O. Etzioni, H. Kautz, “Towards a theory of natural language interfaces to databases” in 8th International Conference on Intelligent user interfaces, Miami, 2003H

[7] A.M. BHADGALE, S.R. GAVAS, M.M. PATIL, and P.R. GOYAL, “NATURAL LANGUAGE TO SQL CONVERSION SYSTEM” in International Journal of Computer Science, 2013

[8] W.A. Woods, R.M. Kaplan, and B.N. Webber, “The Lunar Sciences Natural Language Information System”, Final Report, vol. BBN Report 2378, Bolt Beranek and Newman Inc., Cambridg'e, Massachusetts, 1972

[9] W.A. Woods, “An experimental parsing system for transition network grammars. In Natural language Processing”, R. Rustin, Ed.,Algorithmic Press, New York, 1973

[10] G.G. Hendrix, E.D. Sacerdoti, D. Sagalowicz, J. Slocum, “Developing a natural language interface to complex data”, ACM Transactions on database systems, 3(2), pp. 105- 147, 1978

[11] G. Hendrix, “The LIFER manual A guide to building practical natural language interfaces”, SRI Artificial Intelligence Center, Menlo Park, Calif. Tech. Note 138, 1977

[12] I. Androutsopoulos, “Interfacing a Natural Language Front-End to a Relational Database (MSc thesis)”, Technical paper 11, Dept. of AI, Univ. of Edinburgh, to structured query language(SQL), 1993

[13] A.M. Popescu, A. Armanasu, O. Etzioni, David Ko, and A. Yates, “Modern Natural Language Interfaces to Databases: Composing Statistical Parsing with Semantic Tractability”, COLING\(2004).61

[14] Y.W. Wong, “Learning for Semantic Parsing Using Statistical Machine Translation Techniques”, Technical Report UT-AI-05-323, University of Texas at Austin, Artificial Intelligence Lab, October 2005.

[15] Li. Yunyao, H. Yang, and H. V. Jagadish, “NaLIX: an interactive natural language interface for querying XML”, in Proceedings of the 2005 ACM SIGMOD International conference on Management of data. ACM, 2005

[16] Li. Yunyao, H. Yang, and H. V. Jagadish. “NaLIX: A generic natural language search environment for XML data” ACM Transactions on Database Systems (TODS) 32.4 , 2007

[17] F.A. El-Mouadib, Z.S. Zubi, A.A. Almagrous, and I.S. El-Feghi, “Generic Interactive Natural Language Interface to Databases (GINLIDB)”, in International journal of computers, 2009

[18] J.E. Hopcroft, J.D. Ullman, "Introduction to Automata Theory, Languages, and Computation", Addison-Wesley, Chapter 4: Context-Free Grammars, pp. 77–106; Chapter 6: Properties of Context-Free Languages, pp. 125–137, 1979

[19] D.E. Rumelhart, J.L. McClelland and the PDP Research Group, "Parallel Distributed Processing: Explorations in the Microstructure of Cognition", Volume 1: Foundations, Cambridge, MA: MIT Press,1986

[20] A. Kaur " Punjabi language interface to databases" Thapar university Patiala – 147004 June, 2010

[21] H. Jain, P. Bhatia "Hindi language interface to databases" Thapar university. Patiala – 147004 June, 2011

[22] N. Nihalani, S. Silakari, M. Motwani, "INTELLIGENT QUERY CONVERTER: A DOMAIN INDEPENDENT INTERFACE FOR CONVERSION OF NATURAL LANGUAGE QUERIES IN ENGLISH TO SQL", in International journal of Computer Engineering & Technology (IJCET), 4(2), 379 – 385

[23] A. Kumar, K.S. Vaisla, "Natural Language Interface to Databases: Development Techniques" *Elixir Comp. Sci. & Engg.* 58, 14724-14727, 2013

[24] F. Li and H. V. Jagadish. "NaLIR: an interactive natural language interface for querying relational databases." Proceedings of the 2014 ACM SIGMOD international conference on Management of data. ACM, 2014