International Journal of Computer Architecture and Mobility
(ISSN 2319-9229) Volume 1-Issue 6, April 2013

A Novel Architecture for the Natural Language Interface to Databases

[1]B.Sujatha, [2]Dr.S.Viswanadha Raju, [3]Humera Shaziya

[1]Research Scholar, Dept. of CSE, JNTUH, Hyderabad, AP, India

[2]Professor & Head, Dept. of CSE, J.N.T.University, Jagtial

[3]Lecturer in Computers, Dept. of M.C.A, Nizam College, Hyderabad

Abstract: Aspects of an intelligent interface that provides natural language access to a database is described. The overall system architecture is presented, showing how a user is buffered from the actual database management systems (DBMSs) by three layers of insulating components. These layers operate in series to convert natural language queries into calls to DBMSs. Attention is then focused on the first of the insulating components, the natural language system. A pragmatic approach to language access that has proved useful for building interfaces to databases is described and illustrated by examples. Special language features that increase system usability, such as spelling correction, processing of incomplete inputs, and run-time system personalization, are also discussed. The language system is contrasted with other work in applied natural language processing, and the system's limitations are analyzed.

Key Words and Phrases: natural language, intelligent interface, database access, natural language processing

1. Introduction

Integration of natural language processing (NLP) capabilities (e.g. semantic interpretation of a sentence) with database technologies (e.g. database management systems) has been an interesting task to both the artificial intelligence (AI) community and the database (DB) community since the late 60‟s and early 70‟s. Even now, there is ongoing effort in materializing earlier research goals into practical applications. A central issue is creating natural language interfaces for databases (NLIDBs) so that a user can query a database in a natural language (e.g. English).

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). The example queries is a dialogue

between the user and EFlex a research-based NLIDB the system's responses are similar to the outputs produced by any commercially available DBMS system. Access to information in a fast and reliable way is very important for modern society. Natural Language Interfaces to Databases (NLIDBs) permit users to formulate queries in natural language, providing access to information without requiring knowledge of programming or database query languages. However, despite the achievements attained in this area, present day NLIDBs do not guarantee correct translation of natural language queries into database languages. Moreover, queries are limited to the database domain configured by the database administrator. In a Survey on the importance of natural language processing (NLP) systems, Large User Group" professional society, mentions that:(1) NLIDBs are the most useful application for organizations among all NLP systems, (2) The five most desirable capacities of NLIDBs are: efficiency, domain independence, pronoun handling, understanding of elliptical entries (i.e., implied words), and processing of sentences with complex nouns, and (3) 50% of the best NLIDBs are those that offer domain independence. This paper describes an approach that uses database dictionary. This allows for translation of queries expressed in natural language (English in our case) into SQL, with easy adaptation to different domains. There has been much work on NLP recently, but the area has been around for a relatively long time in the computing world. The main aim of NLP research is to create a better interface to the computer. Spoken language is the most natural interface available for humans to use, but computers are still unable to come close to the rich communication humans can achieve with each other.

Science fiction has created many robots or computers that are able to understand and carry out tasks based on spoken orders or communication. 'Data', an android from the *Star Trek the Next Generation* movies and series can communicate as well as any human in English. The 'HAL' computer from the book and movie *2001 a Space Odyssey* converses verbally with the members of the space ship. Even a toaster from the book and television series *Red Dwarf* manages to hold an intelligent conversation. As these authors have been imagining computers that communicate with humans through natural language, computer scientists have been attempting to make it a reality, but success has so far been limited to specific domains. Here are some examples of early NLP systems:

- ELIZA – by Joseph Weisenbaum (1966). This program is a natural language interface to a psychiatrist. It used pattern-matching rules that were triggered based on key words found in user's dialog. ELIZA used literal text form within users dialog to reformulate questions. There was no 'understanding' of what was being said, ELIZA just gave back questions that seemed most relevant according to the last user input. Weisenbaum reported that some subjects were convinced that ELIZA was a real person. He notes "The human speaker will contribute much to clothe ELIZA's responses investments of plausibility."
- SHRDLU – by Terry Winnograd (1973). This is one of the first programs that could carry out tasks and provide responses in natural language well. It was bound within an artificial blocks world of coloured bricks and pyramids. SHRDLU was able to perform tasks like moving objects around within the limited world, when directed to do so in English. The program used a procedural representation for semantics. This means that each English predicate or term was associated with a procedure that conveyed the meaning (or semantics) of the term. The problem with procedural semantics is that they do not scale up into large domains. Today there is much more demand for better interfaces to computers and much has been achieved in areas of Graphical User Interface (GUI) development. The Windows and Macintosh operating systems are both based primarily on a GUI environment. However, NLP systems have not yet become widely used in the computer world. The main reason for this is that good NLP systems are very difficult to make, due to the scale-up problems encountered in large domains. ELIZA got around this problem by reusing the users input to the system to formulate new questions, and SHRDLU was limited to blocks world, so that the domain would be small enough to handle. Ambiguity in English sentences becomes a more important problem when larger domains are considered and no method to resolve this ambiguity correctly has yet been discovered (but many people are trying).

2. System Architecture

The NLIDB system is being developed for the English language. It has additional elements with respect to other similar systems, better language coverage, much better portability of DBMS and operating system, transparent access through Internet, and a greater use of easiness. The three-level client intermediate-server structure is used, The functionality of each level has been defined. The client is much simpler, which partially solves the problems of the current QBE interface, at the cost of a more active role of the intermediary level. The architecture of the system is shown in Figure 1. The user interface for client is coded in Java Swing .The interface present to the user and user is allowed to enter

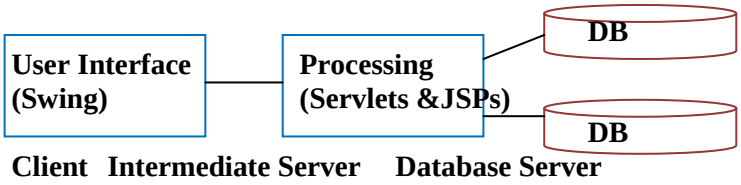

Fig 1: System Architecture

The query is entered in a text field provided to the user on the user interface. Sometime it is difficult for the user to frame a query, and in such situation the user can refer to the list of words or synonyms that are contained in the dictionary provided to the user through the user interface. In cases where users want to enter his own word as a synonym to the existing keywords, he can do so by entering the synonym for the word in the dictionary. Most of the times, it is difficult for the inexperienced users and which also lack a lot of very important semantic information to understand the query framing and for that purpose few example queries will be provided to the user in the interface. The presentation of this example queries and dictionary permits the user to better understand the contents of the database, which facilitates query formulation. The table names are presented on the interface that helps the user to find out what are the tables present in the database.

The user's query is first converted into the SQL query by the Natural Language component of the system.

The SQL query is then forwarded to the DBMS which understand it directly and execute it and send the result

to the intermediate server, which then forward it to the user interface.

The architecture of the system is designed so that the processing takes less time and completes the query execution fast. The three-level architecture consists of client level, intermediate server level and the database level. The client level presents a user interface to the user which consists of several components. The components are – a text field for the user to enter a natural language query named query there are two buttons i) execute ii) clear .To execute the natural query user has to click on the execute button and the clear button is used to clear the contents of the text field, a status field that shows the status of the execution of the query, result is displayed on the result field. The other two fields are database details and the table details field. To connect to the database the following details have to be provided by the user-database name, host name, user name and schema name. After entering the database details the tables present in the database are listed in the table field, so that it is convenient for the user to look up for the tables contained in the database. The processing is performed in the intermediate server that will be implemented in the servlets and JSPs. The intermediate server contains a component called as natural language component that actually translate the query from natural language to the SQL query and the component will be implemented in the java language. The database server can be mysql or oracle and further more database servers will be added as the system progresses forward.

and from an "expert" that contributes with the information that is not usually present in the data dictionary. This module of NLP forwards the natural language to the module of the intermediary's session, which returns it to the client. Finally, it is formatted for final presentation to the user.

After the user connects to the database and is presented the information on it through dictionary, he/she introduces a query using the text interface. The output of this interface is received by the client and passed on to the module of the intermediary's session, which passes it to the natural language processing module (NLP). The architecture of the Natural Language Component of the system is designed in such a manner that it performs the conversion process efficiently. When it finds an error it reports the error to the user and SQL query will not be generated. Syntactic analyzer will look up for the syntax in the query. The semantic analyzer will determine the meaning of the parts of the query. The Code generator will generate the code ie and SQL query from the natural language query.

3. Natural Language Component

Natural Language Component of the EFlex System performs the translation of the natural language to SQL query. To be able to introduce the ontology to the user, the client communicates with the intermediary module. The latter generates a session thread that forwards the query to a module that builds the SQL query from the information of the data dictionary.This significantly helps in the processing of natural language. At the same time, a data dictionary holds information of the attributes, entities, and views that constitute the database, as well as the relationships between the attributes and entities, and even some semantic information that, similarly to the lexicon, can substantially improve the processing of natural language.

As an example, consider the query "what is the income of the of John Smith." The lexicon would indicate that the query is correct, but the information of the data dictionary represented through tables would facilitate detection of possible anomalies at early stages of the analysis. In this example, the requested data is not defined as an attribute. However, the user can consult the information on synonyms without the necessity for them to be defined in the tables, since the synonyms are usually defined in the lexical dictionaries. of the NLP module is quite standard, except that a module that acts between the data dictionary and the NLP lexicon is added, as shown in Fig. 2.

The NLP module receives a query in natural language from the session module and returns it

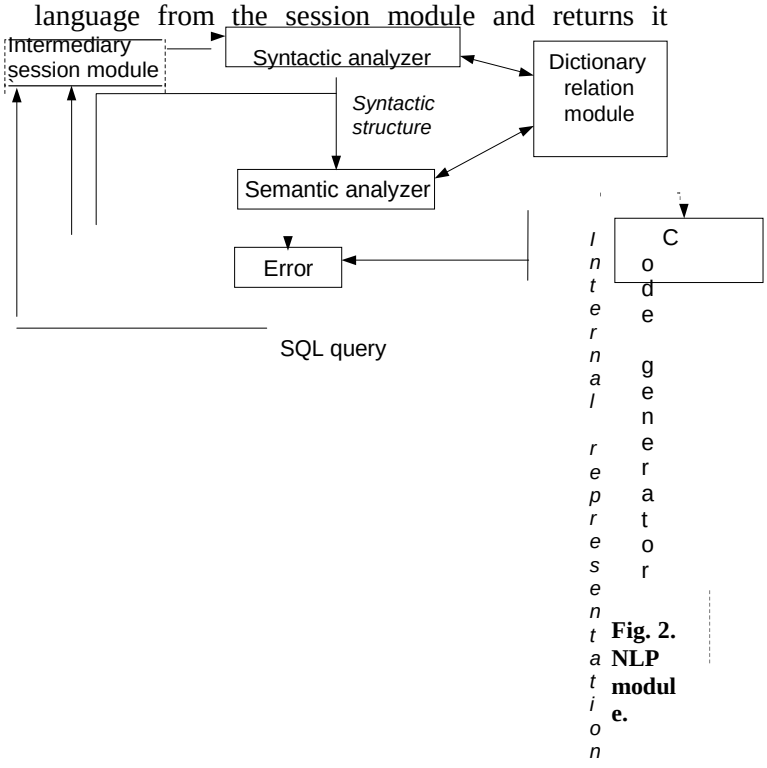

Fig. 2. NLP module.

3.2 Example queries

 # SELECT * FROM table_name

> show me all employee

SELECT * FROM employee

> tell me all our employee

SELECT * FROM employee

> display all employee

SELECT * FROM employee

> list all our employee

SELECT * FROM employee

> Who are our employees?

SELECT * FROM employee

> What is our employee?

SELECT * FROM employee

 # SELECT DISTINCT field FROM table

> tell me our employee salary

What are your trying to say?

> display employee numbers

SELECT DISTINCT employee_number FROM employee

> list all our employee salary

SELECT DISTINCT salary FROM employee

> what are our employee names?

SELECT DISTINCT employee_name FROM employee

> show me the department_number of the employee

SELECT DISTINCT department_number FROM employee

> what are the total salaries of our employee?

Could you rephrase that?

 # SELECT DISTINCT field1, field2 FROM table

> tell me our employee_names and employee_numbers

SELECT DISTINCT employee_name, employee_number FROM employee

> what are our employee salaries and total payments?

What are your trying to say?

SELECT * FROM table WHERE condition

> show me all suppliers whose names are "ABC XYZ"

SELECT * FROM employee WHERE name = "abc xyz"

> tell me our employee with the job starts with AN

SELECT * FROM employee WHERE job LIKE "AN"

> display all employee whose salary between 1000 and 2000

SELECT * FROM employee WHERE salary >= 1000 AND salary <= 2000

> list all our employee whose salary is lower than 2500

SELECT * FROM employee WHERE salary <2500

 # SELECT DISTINCT field1 FROM table WHERE condition

 # SELECT DISTINCT field1, field2 FROM table WHERE condition

COUNT QUERIES

SELECT COUNT(*) AS number_of_table FROM table

> How many suppliers are there?

SELECT COUNT(*) FROM EMPLOYEE

 # SELECT COUNT(*) AS number_of_table FROM table WHERE WHERE condition

>How many departments are there by the number of 20

SELECT COUNT(*) AS number_of_employee

FROM employee

WHERE department_number = 20

> how many employee_ names starts with a

SELECT COUNT(*) AS number_of_employee

FROM employee

WHERE employee_name LIKE "a%"

SUM QUERIES

 # SELECT SUM(field) AS total_field FROM table

> show me the total salaries of employee

SELECT SUM(salary) AS total_cost FROM employee

 # SELECT SUM(field1) AS total_field FROM table WHERE condition

> what is the total salaries of employee with department_number 10?

SELECT SUM(salary) AS total_salary

FROM employee

WHERE department_number = 10

AVERAGE QUERIES

SELECT AVG(field) AS average_of_field FROM table

> show me the average salary of our employee

SELECT AVG(salary) AS average_salary

FROM employee

 # SELECT AVG(field1) AS average_of_field1 FROM table WHERE condition

5. Conclusion

To develop natural language interfaces is very important because of the necessity of providing access to computers to all members of society. With NLIs the access language to computers will be the same that people use, either in written or spoken form.

A study carried out by a group of information system administrators on the usefulness of different applications of natural language interfaces concluded that those used for obtaining information from databases was preferred by users over those of information retrieval and text preparation. This type of interfaces left very far behind other applications such as language translation.

There are many aspects in natural language processing to work on, such as linguistics, computational linguistic, psychology, psycholinguistics, etc. In India, though there is some work related to NLP, very few projects deal with database querying.

The work on natural language interfaces is necessary because there are more and more people that need access to computer resources but do not have experience in this nor usually time to acquire it.

7. References

[1] I. Androutsopoulos, G.D. Ritchie, P. Thanisch, *Natural Language Interfaces to Databases, An Introduction*, citeseer.nj.nec.com/1natural.html.

[2] I. Androutsopoulos, G. Ritchie, and P. Thanisch, An Efficient and Portable Natural Language Query Interface for Relational Databases, *6th Intern. Conf. on Industrial & Engineering Applications of Artificial Intelligence and Expert Systems,* Edinburgh, U.K. Gordon and Breach Publishers Inc., Langhorne, US, 1993.

[3] I. Androutsopoulos, *A Principled Framework For Constructing Natural language Interfaces to Temporal Databases,* PhD thesis, research paper No.709, Dept. of Artificial Intelligence, Edinburgh University, Scotland, UK. 1996.

[4] P. Auxerre, *MASQUE Modular Answering System for Queries in English, Programer's Manual*, technical report AIAI/SR/11, Artificial Intelligence Applications Institute, University of Edinburgh, March 1986.

[5] P. Auxerre and R. Inder, *MASQUE Modular Answering System for Queries in English, User's Manual*, technical report AIAI/SR/10, Artificial Intelligence Applications Institute, University of Edinburgh, June 1986.

[6] BBN Systems and Technologies, *BBN Parlance Interface Software – System Overview*, 1989.

[7] M. Bates, M.G. Moser, and D. Stallard, The IRUS transportable natural language database interface, in L. Kerschberg(Ed.),*Expert Database Systems*, Benjamin/Cummings, 1986.

[8] P. Bernstein, M. Brodie, S. Ceri, D. De Witt, M.Franklin, H. García-Molina, J. Gray, J. Held, J. Ellerstein, H.V.Jagadish, M. Lesk, D. Maier, J. Naughton, H. Pirahesh, M. Stonebraker, and J. Ullman, *The Asilomar Report on Database Research* , 1998.

[9] R.J. Bobrow, The RUS System, *Research in Natural Language nderstanding*, BBN report 3878. Bolt Beranek and Newman Inc., Cam- bridge, Massachusetts, 1978.

[10] G. Carreón Valdés, *Herramienta para Consultas EzQ para Multibases de Datos en Internet*, M.S. thesis, CENIDET, Cuernavaca, Mexico.

[11] N. Cercone, P. McFetridge, F, Popowich, D. Fass, C. Groeneboer, G. Hall, *The SystemX Natural Languaje Interface: Design, Implementation and Evaluation*, Centre for Systerns Science, Simon Fraser University, British Columbia, Nov. 1993.

[12] E.F. Codd. A Relational Model for Large Shared Data Banks, *Comm. of the ACM*, 13(6), 1970.

[13] E.F. Codd. Seven Steps to Rendezvous with the Casual User, in J. Kimbie and K. Koffeman (Eds.), *Data Base Management*, North•Holland Publishers, 1974.

[14] E. Chay Coyoc, *Una Interfaz en Lenguaje Natu- ral en Español para Consultas a Bases de Datos*, MS thesis, ITESM-Cuernavaca, Mexico,1990.
[15] S.M. Dekleva, Is Natural Language Querying Practical? *Data Base*, May 1994.
[16] B.J. Grosz. TEAM: A Transportable Natural- Language Interface System, *1st Conf. on Applied Natural Language Processing*, SantaMonica, California, 1983.
[17] B.J. Grosz, D.E. Appelt, P.A. Martin, and F.C.N. Pereira, TEAM: An Experiment in the Design of Transportable Natural•Language Interfaces, *Artificial Intelligence*, 32, 1987.
[18] C.D Hafner and K. Godden, Portability of Syn- tax and Semantics in Datalog, *ACM Transactions on Office Information Systems*, 3(2), 1985. [19] G. Hendrix, E. Sacerdoti, D. Sagalowicz, and J.Slocum,Developing a Natural Language Interface to Complex Data, *ACM Transactions on Database Systems*, 3(2), 1978.
[20] L.R. Harris, The ROBOT System: Natural Language Processing Applied to Data Base Query, *ACM'78 Annual Conference*, 1978.
[21] V. O. Huerta H., *Un Métodopara el Reconocimiento a Bases de Datos en Interroga- ciones en Lenguaje Natural*, MS thesis, ITESMCuernavaca, Mexico.
[22] International Business Machines; http://www-4.ibm.com/software/speech/es/.
[23] H. Jiménez S., *Adaptación de Algoritmos deAprendizaje Mecánicopara Obtención de Reglas en Corpora del Español de México*, PhD thesis, BUAP, Puebla, Mexico.
[24] R. Kasper, A Flexible Interface for Linking Applications to Penman's Sentence Generator, *DARPA Speech and Natural Language Work- shop*, 1989.
[25] A. May Arrioja, *Herramienta para Consultas Basadas en Ejemplos (QBE) para Multibases de Datos en Internet*, MS thesis, CENIDET, Cuernavaca, Mexico, 2000.
[26] F. Marcos Marín, A. Moreno, C. Olmeda, J.Martínez, and S. Guilarte. Proyecto Sylvia; www.lllf.uam.es/proyectos/sylvia.html.
[27] R. J. Money, *Inductive Logic Programing for Natural Language Processing*, Dept. ComputerSciences, Univ. of Texas.
[28] F. Rasgado Celaya, *Herramienta para Consultas Basadas en Ejemplos (QBE) para una Base de Datos en Internet*, MS thesis, CENIDET, Cuernavaca, Mexico, 1999.
[29] P. Reis, J. M. Nuno Mamede, *Edite – A Natural Language Interface to Databases: a New Dimension for an Old Approach.* INESC, Portugal.
[30] P. Resnik. *Access to Multiple Underlying Systems in JANUS*, BBN report 7142, Bolt Beranek and Newman Inc., Cambridge, 1989.
[31] G. R. Rocher Silva, *Traducción de Queries en Prolog a SQL*, BE thesis, Escuela de Ingeniería, Universidad de las Américas-Puebla, 1999.
[32] J. Rojas, J. Torres, A Survey in Natural Language Databases Interfaces, *8vo. Congreso Intern. de Investigación en Ciencias Computa- cionales*, Inst. Tecnol. de Colima, Mexico 2001.
[33] M. Ruiz, A. Diekema, P. Sheridan, *CINDOR Conceptual Interlingua Document Retrieval:TREC-8 Evaluation*, MNIS-TextWise Labs, Syracuse, NY 13202.
[34] R.J.H. Scha, Philips Question Answering System PHILIQA1, *SIGART Newsletter*, No. 61. ACM, New York, 1977.
[35] V. Sethi, *Natural Language Interfaces to Databases: MIS Impact, and a Survey of Their Use and Importance*, University of Pittsburgh, PA, USA.
[36] DCC, *SiMBaDD Sistema Manejador de Basesde Datos Distribuidas*, CENIDET, Cuernavaca, Mexico, www.sd-cenidet.com.mx/simbadd.
[37] D. Stumberger, B. Ballard, Semantic Acquisition in TELI, *24th Annual Meeting of ACL*, New York, 1986.
[38] M. Templeton and J. Burger, Problems in Natural Language Interface to DBMS with Examples from EUFID, *1st Conference on Applied Natural Language Processing*, Santa Monica, California, 1983.
[39] B.H. Thompson and F.B. Thompson, Introducing ASK, A Simple Knowledgeable System, *1st Conference on*

Applied NaturalLanguage Processing, Santa Monica, CA, 1983.

[40] B.H. Thompson, C. Raymond, and J. Money, *Automatic Construction of Semantic Lexicons for Learning Natural Language Interfaces*, Stanford Univ., University of Texas.

[41] Trinzic Corporation, Bethesda, MD *INTELLECT–Natural Language System.* (commercial triptych).

[42] D.L. Waltz, An English Language Question Answering System for a Large Relational Database, *Comm. of the ACM*, 21(7), 1978.

[43] D. Warren and F. Pereira, An Efficient Easily Adaptable System for Interpreting Natural Lan- guage Queries, *Comp.Linguistics*, 8 (3-4), 1978.

[44] A. W. Penn, *DB Valet: A natural language database interface*, MS thesis, Univ. of Lousville, 2000, www.louisville.edu.

[45] W.A. Woods, Procedural Semantics for a Question-Answering Machine *Fall Joint Computer Conference*, NY, 1968. AFIPS.