

Using Natural Language Processing in Order to Create SQL Queries

F.Siasar djahantighi¹, M.Norouzifard¹, S.H.Davarpanah², M.H.Shenassa³

¹Islamic Azad University science and research Branch, Tehran, Iran

²University of Science and Culture, Iran

³K.N.Toosi University of Technology, Iran

Email (F.Siasar@Gmail.com, Mnorouzif@yahoo.com, Davarpanah@usc.ac.ir, Shenassa@eetd.kntu.ac.ir)

Abstract

Using query language for dealing with databases is always a professional and complex problem. This complexity causes the user's usage of data existing in database limits to use definite reports there are in some pre implemented softwares. However, you can create this opportunity that each none professional user transfers his questions and requirements to computer in natural language and derives his desired data by natural language processing. In this paper we represent a method for building a "Natural Languages Interfaces to Data Bases" (NLIDB) system. This system prepares an "expert system" implemented in prolog which it can identify synonymous words in any language. It first parses the input sentences, and then the natural language expressions are transformed to SQL language.

I. INTRODUCTION

Nowadays there are too many data which maintain in organizations, companies and universities databases, but only the individuals who are familiar with data query methods can directly use these data. It is clear that if people can ask their question in natural language then the desired data prepare, the process will continue faster and with higher quality.

Some methods and softwares such as query by example softwares have been designed, but these softwares do not have the ability to create a complex query.

To solve this problem, some commercial softwares have been represented[1] that execute lexical pars and semantic analysis on natural language sentences with natural language processing and then transform them to SQL query language commands, which it can produce the user's data from database. The main problem of the prior methods and softwares was in the semantic

analysis when they emit more than one output for a single input. This problem has been solved by considering an expert system. Different steps of these systems are shown in Figure 1.

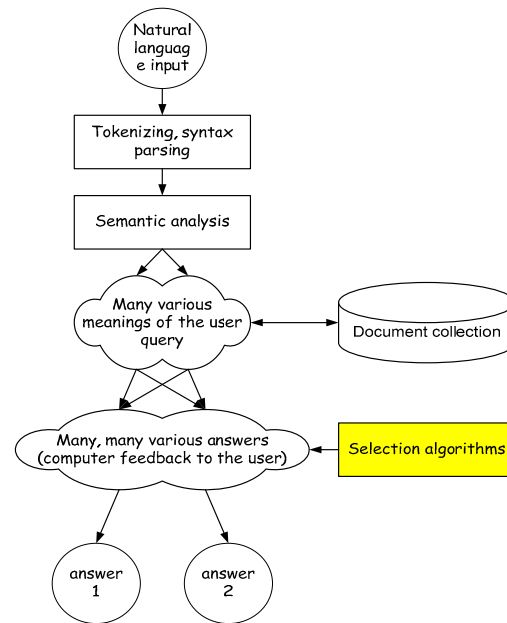


Figure 1. Question answering system

II. SIMILAR TENDENCY

In 1999, for the first time, a research group named cognitive computation group used superficial parsing method. The first software in this domain presented in November 1999[1].

Same as Natural Languages Interfaces to Data Bases (NLIDB) other systems have been surveyed, the result of these surveys was similar to natural language processing systems and successfully implement the natural language processing. Such as:

The activities implement for communicate between natural language and programming. With their method we can act as like as a programmer, without any knowledge about programming languages, just by using “natural language”. This is very similar to “Natural Languages Interfaces to Data Bases”[2].

Another similar system is question answering system which is a branch of natural language processing. In this system user asks his question in natural language and receives the appropriate answer.

Question answering system (QA) tries to answer natural language question by analyzing an infinite and unstructured set of texts. While Natural Languages Interfaces to Data Bases (NLIDB) deals with structured text which has been parsed and its entities and attributes has been identified before[3].

You can discover the difference between prior systems algorithm and the presented method by comparing Figure 2.

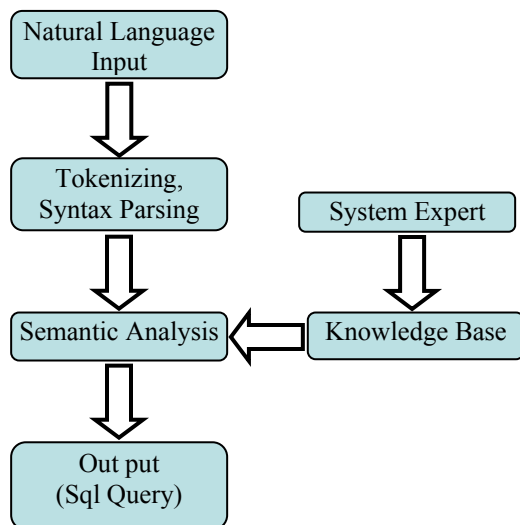


Figure 2. The flowchart of the purposed algorithm.

NLIDB is a system that allows a user to interact with a database by expressing himself in a natural language such as English or Persian or any other natural language[3].

“Natural Languages Interfaces to Data Bases” (NLIDB) needs a mixture of syntactic knowledge and semantic knowledge[4].

III. UNDERSTANDING NATURAL LANGUAGE BY MACHINE

A. Syntactic knowledge

Restoring data in natural language processing, by syntactic viewpoint includes following steps [5] :

- Shallow parsing
- Syntactic parsing
- Lexical cohesion

Shallow parsing. Shallow parsing is used in restoring information; in this method some functions are used to score the text. These functions give higher scores to “main structures” that often use in input sentences.

Syntactic parsing. This step because of multiplicity of sentences which must be parsed in this step takes too times. Complete pars includes considering the number of whole words and many of “main structures” in natural language (an example of these main structures has been showed in Table 1). This parsing can not be easily done because the possibility of confronting with problems in deriving this information is too high.

Table 2 includes some of “main structures” of natural language sentences. The parser with these identified main structures tries to characterize the possible role for each single word of input text. Sometimes the input sentence is adapted with some main structures of language. Every structure specifies the role of different words. In fact, finding some “main structures” adapted with input sentence is the recognition of different roles for words of input sentence by parser.

One of the most common way to use all information existing in pars tree to restore them is exploring behavior of basic component of the sentence. We can specify the type of existing words in subsequent sentences more easily by detaching main component of a sentence and characterizing their type according to the text (i.e. Noun phrase(NP), verb phrase(VP),...).

NOFUNC	NP	NPSBJ	VPS	PP
ADVP	VP	ADJPPRD	NPPRD	VPSBAR
NPTMP	ADVPTMP	VPSTPC	VPSNOM	NPLGS
SBARADV	ADJP	NPADV	VPSADV	VPSINV
VPSPRP	ADVPMNR	SBARTMP	PPPRP	
PPLOCCLR	SBARPRP		PPPRD	ADVPCLR
VPSPRN	VPSCLR	NPLOC	ADVLOC	ADVDIR
PPDTV	ADVPPRD	WHNP		CONJP
NPHLN	VPSQ	VPSNOMSBJ	SBARPRD	VPSPRD
NPCLR	PPPUT	NPTTL	ADJPPRDS	NPTMPCLR
	INTJ		PPTMPCLR	PPCLR

TABLE 3: SYNTACTIC PARSING TABLE

Lexical cohesion. For implementing an appropriate parsing, we must consider the lexical cohesion. It is possible that there be some multi word expressions that have different meaning beside each others. These words beside each others constitute an entry of dictionary. If each of them be used separately, it will

be possible they transmit another meaning. This kind of expressions called “cohesive lexical units”.

For example, the expression “fire hydrant” contains different words that each one have its own meaning but the dictionary shows that by putting them beside each others it means different, and according to the text type and other used words in that text we can get an appropriate understanding from “cohesive lexical units”. However this subject is a part of natural language processing that has not been identified exactly and yet we have not found any proper solution. The researches represented that using of variety and miscellaneous information about theory methods for characterizing cohesive lexical units could improve the performance of restoring[5].

B. Semantic knowledge

Natural Languages Interfaces to Data Bases (NLIDB) needs a preprocessor to realize the changing of the words in input sentence. This preprocessor creates a semantic database from different kinds of expository rules of the language and semantic sets for all entities and all their possible attributes.

This database helps us to present an equal query for different sentences. As like as following sentences:[3]

Who is (are) the author(s) of the book(s) “Natural language processing of prolog”?

Who is (are) the writer(s) of the book(s) “Natural language processing of prolog”?

Who is (are) the author(s) of the resource(s) “Natural language processing of prolog”?

Who is (are) the writer(s) of the resource(s) “Natural language processing of prolog”?

To create this semantic database the preprocessor reads the schema of database and identifies all of entities and their attributes and finally creates a list of synonymous and similar words with WordNet[6]. Then, this primeval semantic set exposures an expert individual for editing to desired subject and domain. While running this set, it is used to find the most similar entity name to the terms of input sentence.

For example, in semantic database, the preprocessor creates a semantic set including the words {writer, author, generator, source, communicator, person, individual, maker, shaper, coauthor, novelist,...} for an entity named author, then the expert individual limits them to { author, creator, generator, writer }. So with these semantic sets, each of our four sentences creates similar SQL query, because they use synonymous words, in other words they there are in same semantic set.

For better understand, lets take a look at following example[3]:

Suppose there are three entities named resource author, resource and author. These three entities and their relationships are shown in **Error! Reference source not found.**

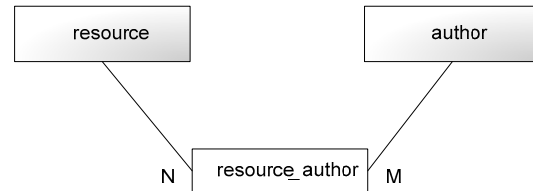


Figure 3. An example of resource, author, and resource_author entities.

Suppose that resource entity has id ‘title and publish date attributes. Author entity has id ‘name and address attribute and finally resource author entity has two foreign keys to other entities.

With this small schema, the preprocessor characterizes the semantic set for resource entity with WordNet. The expert individual directs the preprocessor where to accept {book, document, article, paper, report, resource, thesis, and tutorial} as the possible reference for resource. If we can’t find any similar word for this database with WordNet, the expert individual must specify subject and domain, so preprocessor can create necessary semantic set. For example, for resource author entity that will not be found in words list of WordNet, the expert individual can specify similar words, so preprocessor can create proper semantic set.

In this paper we use an expert system instead of expert individual. Table 5 shows deference’s between result of presented system, with using expert individual and expert system.

C. Expert System, Prolog, and Amzi

An expert system is a program that acts like an expert individual in some domain[7].

In expert systems making decision is based on existing knowledge in expert system knowledge base. An expert system explores the knowledge and finally decides definitely or indefinitely. Such systems include two main sections:

- Knowledge base
- Inference engine [4][8] (see FIGURE 3)

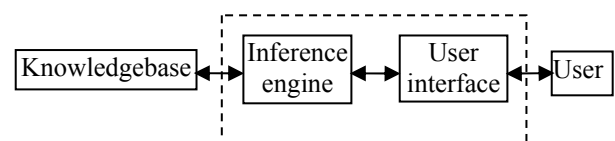


Figure 3. Expert system structure

Prolog is an abbreviation for “programming in logic”. This language is particular for responding to the questions of a knowledgebase that has structured rules and reality. Prolog, in itself, has backward chaining module and also use another tactic called “reversible detection”[9].

AMZI! LOGIC SERVER. Amzi! Logic server is an interface for the programs which are designed in prolog, and make them presentable and usable in other environments and other programming languages. In result we can create applications, that they use advantages of prolog’s logic rules.

Amzi core is a running engine which executes compiled programs of prolog. Logic server is a dynamic library which contains the running engine. We can connect this library to other applications. This library implemented in c++[10].

D. Semantic analysis

By now, we identify the words which are possibly related to specific entities, and put them in a semantic set. But the preprocessor must identify the words which are possibly related to the attribute of specific entities and, as mentioned, find similar word for them by WordNet, and finally an expert individual or an expert system edit them for special purpose in domain.

One single attribute sets as the default attribute. This attribute will be used to create a query.

For denoting the relationship between entities and characterizing generic pattern for mapping between user’s input and SQL queries, we use particular patterns[3]. Three patterns are shown bas follow:

<attribute> of <object>
 <attribute> of <object> of <object>
 <action verb> object <attribute value>

	Sql Query (out put)				
Expert system	Middle	High	Middle	Middle	High
Expert individual	A little low	High	A little low	Middle	high

The system tests all pars trees created by link parser until it find one pars tree that is adapted with one of these patterns. Then, it searches the pars tree for nouns instead of <object> to replace them with relevant expression. We can consider any kind of strings as a replacement for <attribute>. Finally, it searches the verb phrases in pars tree for <attribute value>s.

Hence, according to variety of pars trees which have been created by parser and possibility of adjustment between them and mentioned patterns, it is possible that we acquire multiple queries.

For example, we spot following SQL query pattern, for the first kind of our patterns:

*SELECT attribute FROM entity1 WHERE
 Default attribute = <value of
 entity1.default_attribute>*

When the system receive a question such as:
 “List the address of the writer Mark Twain”

First of all it will create the pars tree. In this pars tree, list is identified as a verb, address is identified as a noun and other word types are identified. The system’s semantic sets which are created by preprocessor maps writer to author entity and identifies address as the address attribute of the author entity. In the next step mark twain is mapped to default attribute. This attribute is name. Finally the following SQL query will engender:

*SELECT address
 FROM author
 Attribute Relation WHERE name = Mark Twain*

E. Implementing proposal determinant system

To design automatic system for identifying Natural Languages Interfaces to Data Bases, a rule base knowledgebase and a backward chaining inference mechanism used. We use MS-Access for preserving similar words and natural language methods, prolog for inference engine, and VB.NET software environment for designing user interface. In addition, we use the Amzi company interface for communicating between prolog and VB.NET.

This movement between methods is shown as a tree of methods. The created tree for each method presented. Some expressions with prolog, as the rules and realities in knowledgebase, presented.

Providing appropriate viewed, act of movement in decision tree will stop. You can some results of system about this question:

“Who is (are) the author(s) of the book(s) “Natural language processing of prolog”? ”, from 8 issues in the following table: (see Table 4)

TABLE 5: DEFERENCE’S BETWIN EXPERT SYSTEM AND EXPERT INDIVIDUAL

So, by continuing semantic analysis, intelligent determinant system is designed and is implemented. This system explores information based on the existing data on expert system’s knowledgebase. Finally this expert system can exert limitations of a semantic set of word as an expert individual.

IV. CONCLUSION

By providing an expert system, we are encoding hidden mystery of natural language; The fact that common words tend to have multiple meanings can lead to ambiguity, the expert system can maintains database that represents the state of the world by looking at the context surrounding the sentences and receives the best recognized from the text. We collect the required knowledge for this system from an individual who is experienced in natural language analysis, and embed this knowledge into an expert system as a knowledge base. It finds the most similar entity name to the terms of input sentence based on searching this knowledge base. This paper is presenting the result of using an expert system beside common existing solutions for transforming natural language expressions to SQL query language. Result shows this process can be completely automated. In the future, to complete this process, an image processing system will be used to detect questions and sentences automatically. Another knowledge base will be produced as well to handle Persian language.

REFERENCES

- [1] <http://www.vbelf.com/>
- [2] D. Vadas, and J. R. Curran, "Programming with Unrestricted Natural Language", School of Information Technologies, University of Sydney NSW, Australia, 2006.
- [3] N. Stratica, L. Kosseim, and B. C. Desai: Using Semantic Templates for a Natural Language Interface to the CINDI virtual library", Department of Computer Science, Concordia University, Montreal, Canada, 2004.
- [4] M. Sanderson, "Artificial intelligence & natural language processing", Porto, 2000.
- [5] B. E. Lambert, "Improving information retrieval with natural language processing", University of Massachusetts Amherst, USA, 2003.
- [6] G. Miller, "WordNet: a lexical database for English, Princeton University", Princeton, New Jersey, USA, 1995.
- [7] I. Bratko, "Artificial Intelligence and Programming in Logic", pp 323, 2002.
- [8] S. H. Davarpanah, M. Saniei, and M. R. Kangavari, "A Novel Method to Fault Detection in Industrial Systems Using an Adaptive Expert System", The First Iranian Mechatronic Conference, ICME, May 27-28, 2003.
- [9] R. Lovian, R. Drang, B. Adleson, and K. badie "Guide Artificial Intelligence and System Expert to Programming Language C", pp 264, 1995.
- [10] <http://www.Amzi.com/>