International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 8 (2018) pp. 5792-5795

Data Analysıs of Natural Language Queryıng Usıng NLP Interface

Sharath Chander P.[1], J. Soundarya[1], R. Priyadharsini[1] and B. Bharathi[1]

[1] *Department of Computer Science and Engineering, SSN College of Engineering, Kalavakkam, India.*

Abstract

Data is one of the buzz words revolving around any new computational creation. Such data is to be related, manipulated and queried. Structured Query Language(SQL) is used to establish relational model with the data and retrieve necessary facts from the database. Non-experts will find it very difficult to interact and fetch information from the database as it is necessary to have knowledge about SQL. Recalling the keywords and the syntactic rules for joining the various tables in a relational database is another problem. Cases where the name of the attributeis referred in the root database may not be known or visible to the end user. Under such circumstances it is impossible to fetch the information even from a small single relation which leads to the failure of data extraction. This project aims in solving this issue by incorporating natural language called the Natural Language Interface Relational Database System (NLIRDS). It combines the features of the Artificial Intelligence with the Relational Database (RDBMS). Earlier systems were developed where the semantic maps were created manually,whereas in this system it is generated automatically.

Keywords: NLQ (Natural Language Query), NLP(Natural Language processing), MR (Meaningful Representation).

INTRODUCTION

The natural language interface to a database querying tool makes the users to query in his/her own language conveniently which does not force the user to follow any syntax or keywords as in a structured language. This is achieved by IIRS, the system proposed in this project. Additionally, IIRS also handles textese or texting language and convert them into SQL query. For example, the attribute, employee name is named as "e_name" in the database, suppose the user writes a query where he/she can represent the same as "emp_name" or"employee_name" etc. Thus until and unless the exact attribute is known analysis of data would be tedious. IIRS resolves this problem. It allows the users to use any relevant attribute name, keywords and processes them internally to extract the desired query. An example for textese is given as: "wat is d name of city startng with A?". Here 'wat','d', 'startng' are processed into 'what', 'the', and 'starting' respectively. Thus wrongly entered words due to human errors can also be interpreted and corrected automatically.

RELATED WORKS

Database NLP is one of the most important and successful research area ever since it has begun. It dates back to 1973 where LUNAR, a system that answered the queries related to lunar rock samples. LIFER/LADDER is an NLIDB of US navy ships. This system was limited by the fact that it could support only simple one table queries. Avinash J. Agarwal, O.G. Kakde described this semantic analysis using domain ontology[6]. The conversion of NLQ to a meaningful SQL has been proposed earlier [1]. Kaur Saravjeet and Rashmeet Singh Bali had highlighted the conversion of NLQ to internal format based on syntactic and semantic knowledge of natural language. Gauri rao, Snehal Chaudhry, Nikita KulKarni, Dr. S.H. patil. have proposed the translation of NLQ using lexicon implemented in semantic analysis.

WORKİNG

IIRS (Intelligent Information Retrieval System)is the technique being implemented in this work. It overcomes the problems faced by earlier systems. Some of the earlier systems were developed by *suja et al.*

Overview of IIRS

A relational database consists of three entities: attribute name along with its data type, structure of the relational table and relationship among other tables. A question being given or that is entered by the user is first analysed in such a way that all the three entities are satisfied in that question so as to retrieve the information required. To achieve this a semantic map is required where the data elements are represented in a more efficient and easily understood manner. This semantic map is created or developed by using a semantic builder. The semantic builder automatically constructs the semantic map once all the entities are available in the question.

Lexicon

A lexicon is a collection or a set of attributes in a database with all of its possible synonyms represented linguistically. For an example *emp_id* can be represented as *roll_no, reg_no, e_id, serial_no*. Similarly *emp_salary* can be represented as *salary, payment, income, wages.* To build a lexicon the names of the database elements are first extracted and split into separate words. These synonyms are then recognized by using the *wordnet* .For *sms* language interpretation a manually generated wordhouse is created for most frequently used

shortforms for words. It mostly consists of the '**wh**' type words and its shortforms. For example:

1. Wat is interpreted as what

2. whr is interpreted as where

3. whn is interpreted as when

This feature also has another dimension where the question given by the user is ensured error free if any manual misspellings are done.

Relation identifiers

This enables to identify the relationship between tables in the database. The relation is stored in the form *employees.emp_id = salary. emp_id,* where the *employee* table consists of {emp_id, name, DOB, phone number, address}. The *salary* table consists of {emp_id, days_worked, hours_worked, salary}. All these primary works are done before actually analysing the question given by the user. The question which is in natural language without actually involving the syntax and keywords of a regular structured query language has to be processed in multiple levels so as to generate the appropriate output to the end user.

Stages. The multiple stages involve:

SMG (Semantic-MAP generator)

MRG (Meaningful Representation Generation)

Query and result generation

Semantic –Map Generator (SMG)

The input question given by the user is first split into individual tokens which forms the lexicon. The IIRS takes this relational lexicon as the input and in turn generates S-map or a semantic map.

Meaningful Representation Generation (MRG)

The natural language question given by the user is not machine understandable. Hence they must be converted into an intermediate form which is known as the *Meaningful Representation* (MR) . This is done by the IIRS by taking NLQ as the input and generating MR.

In this stage another process called the tokenization happens. This is done with the help of a tokenizer where the NLQ is tokenized into n number of tokens known as the *token set*. For example "List the customer location whose balance is greater than 50,000". This NLQ is tokenized in the following manner {List, the, customer, location, whose, balance, is, greater, than, 50000}This token set is then sent into a *parser*.Parse tree consisting of various tokens and the fitting relationships between them are extracted by using the parser. The fitting relationships among the tokens are then fed into the MR generator which in turn outputs the *MRM (Meaningfully Represented Map).*This map contains information about the NLQ in a *system lucid or conceivable intervening form. M*R Scrutinizer rectifies the MR map and propagates an absolute reformed version.

Query and Result Generatıon

Here the query generator takes in input such as sematic map, MR map and engine. For a given query many possible relevant queries are made available. Theses queries then find their way into the query scrutinizer where proper protocols like the checking for syntax and linguistic correctness is done. Among the many possible queries generated only one is selected that is the *best match* is chosen by the *Limiter.* This gives the final desired output to the user[1]. A dependency resolving function aids in identifying the dependencies among various tokens and exploits the fitting relationship information through a parser.

IIRS theory

Basic definitions

An NLQ is the question put forth by the user to the system in a natural language. For example: show the list of employees whose salary is 10,000. This can be viewed as a set of tokens where NLQ is considered as a string or array of characters.

CA=T1 U T2 U T3 U T4…..Tn.

An ordered set of character array which when combined in an unique way produces the original NLQ. This is called as *tokenization*. An NLQ with atleast one tokenization is said to be tokenizable. A value is represented using rows and columns. A table consists of *attributes* which is the name given to a particular column. Relationship between various tables is called a *relation.* R subset (A X B).Here,

A is the set of columns of table A

B is the set of columns of table B

X is the cross product of A and B.

Semantic amenability

This semantic amenability also known as the '*semantic tractability*' is where a set queries that can be interpreted in a easy and straight forward way. The *semantic amenability model address* selects only the best and easily understandable query from a set of possible queries. In case of complex queries output is produced by converting it into a partially tractable form.For example: *show the details of the youngsters working in the company.* In this query the word youngsters is not specific in its action because the system does not know which age corresponds to the term youngsters. To achieve this the value is made available in the *wordnet*. Now the query would be executed properly.

Pragmatics and enchancements

IIRS focuses on the important parts of the knowledge base and uses its knowledge to overcome the ambiguities and to establish connections among the entities. The input provided in the form of NLQ must be understood by the system. This is achieved with the help of the small knowledge base. For example: in case of subqueries like *to display the students*

who had studied in Harvard university and those of which are currently working for Google. Here the doer is the student and the words *who* and *those of which* also refer to students. In order to establish the link between these words and the subject IIRS uses the knowledgebase where the ambiguities are resolved using the grammar syntax of the language.

IIRS Architecture

IIRS can be visualized as a package embedded with three main components. The components are:

- Semantic building
- MR generation stage
- Query generation stage

Semantic building stage.A RDBMS is an organized set of tables connected to each other with enormous number of data in them. This includes the basic parameters which are nothing but the primary commodities. They are the attribute name, table structure and the relationship between them. The semantic builder performs its work on a *semantic layer.* A *semantic map* for a particular database is created once. It consists of the table information, the relationship and lexicons. The best match for "**l**" with "**name**" as successor is found using the *wordnet.* Finally this can be interpreted as "**last**". When a lot of relevant predictions are present for the '**l'** then the user is requested to pick the desired one manually. In rare cases if the system fails to give out the best match for the lexicon then the user is requested to enter it manually. A *tokenizer* tokenizes the given question. Such tokens can be divided into 3 categories namely, reserved keywords (where, show….etc), attributes and values.This tokenizer provides all possible complete tokenization. Out of these atleast one complete tokenization must map precisely to one set of DB elements (one to one relationship). Each attribute token must correspond to the value token .

MR generation. The highlighting feature of NLIDB is translating a natural language query into *SQL query*. But this cannot be done directly. Hence it is converted into intermediate form i.e MR. MR consists of one or more tokens and the relationships between them i.e the match, DB attribute-attribute token and DB value-value token must be compatible and attached. Each relation token must correspond to an attribute token or value token. Such a mapping that fulfils the above constraints is a valid mapping. IIRS uses an open source NLP or Stanford parser to parse the questions and extract the relationships between the tokens from the parse tree.

Query generation. Query generator outputs the final query by establishing the relationship between the MR map and the semantic information. While mapping the synonyms of the irrelevant words can be found using the wordnet. Eg**:** the set {every, pupil} {entire, seekers} will produce the same amplified set{all, students} in which *students* is the token which will be mapped to the database attribute *student* and *all* is the token which will be considered as the reserved keyword and can be utilized later. In order to avoid ambiguity it is made sure if the other words in the token are also present in the question. For example: *id* can represent multiple database attributes like *student_id, dept_id, hod_id* . But 'id of a student' in the question represents only the *student_id.* In case of SMS language or textese the manually generated list that consists of alternatives is used before parsing through the wordnet. Eg: whr does ram live? Here "whr" finds an alternative in the list i.e whr->where and is replaced.

The query generator makes use of the predefined engine containing reserved keywords and the actions associated with them to knit the final SQL query.

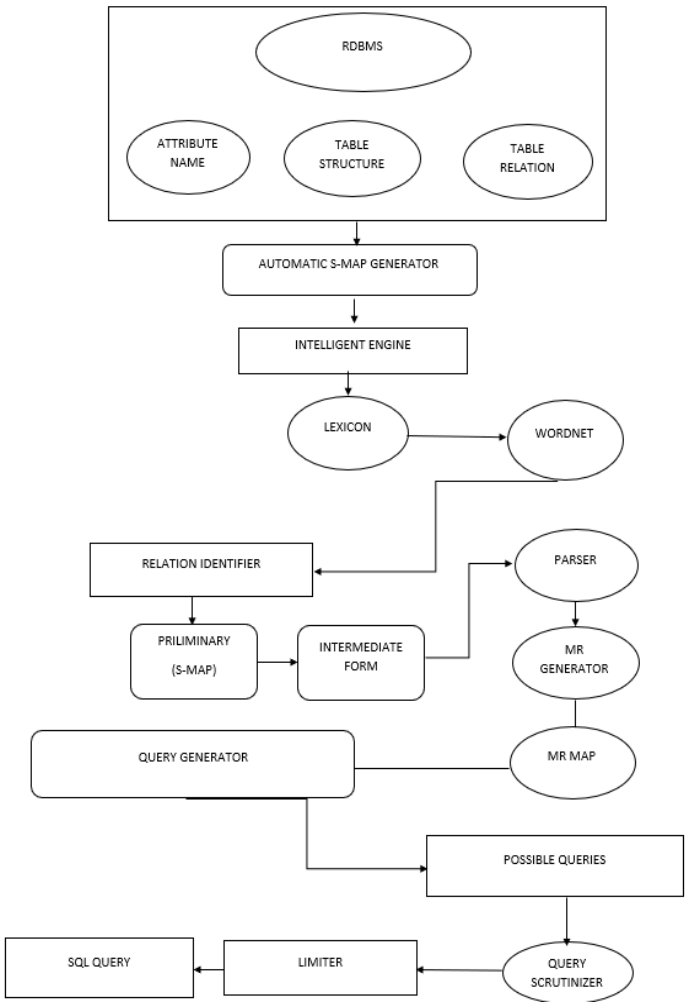

Figure 1. Working of IIRS

CONCLUSION

IIRS serves as an efficient tool in the data analysis. It allows the non experts to handle data conveniently. The presented approach performs concept identification by using event related concepts available in wordnet to find out candidate events from natural language constraints. The formulated output that conveys the information can also be represented in the form of tables, charts and graphs. This system also allows the usage of SMS language making it more user-friendly. The process of lexicon is performed such as lemmatization. Extraction of useful knowledge out of a huge database is

made easy by this system. Future technology involves the speech-voice NLQ wherein the user can fire questions just by his natural spoken language. IIRS in future will find its way in analysing any natural language where NLP is employed in the translation of the natural language to the standard language (English) which in turn is converted into SQL. This makes the extraction of information for the financial operators such as banks, MNC's who have to deal with a large amount of data easily. It appears that even simple data augmentation with synonyms taken from common thesaurus can yield significant improvements for common NLP problems such as sentiment analysis.

REFERENCES

[1] Anil M.Bhagdhale, Sanhita R. Gavas, Meghana M.Patil, Pinki R.Goyal: Natural Language to SQL Conversion System. In: International Journal of Computer Science Engineering and Information Technology Research *(IJCSEITR)* Vol. 3 Issue 2, June, 161- 166(2013)

[2] Prasun Kanthi Ghosh, Saparja Dey, Subhabrata Sengupta: Automatic SQL Query Formation from Natural Language Query. In: International Journal of Computer Applications.

[3] Rohini B.Kokare, Kirti H. Wanjale:A Survey of Natural Language Query Builder Interface to Database. In: International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 5 Issue 4(2015)

[4] F.Siasardjahantighi, M.Norouzifard, S.H. davarpanah, M.H.Shenassa: Using Natural Language Processing in Order to Create SQL Queries. In: Proceedings of the International Conference on Computer and Communication Engineering (2008)

[5] GuiangZangi, Phillip C-Y Sheu:A Natural Language

Database Interface Based on Probabilistic Context Free Grammar. In: *IEEE International workshop* on Semantic Computing and Systems.(2008)

[6] Avinash. J Agarwal,O.G.Kakde: Semantic Analysis of Natural Language Queries Using Domain Ontology for Information Access from Database. In: I. J. intelligent system and applications.12:81-90.(2013)

[7] C. S. Aguilera, D. M. Berry: The Use of a Repeated Phrase Finder in Requirements Extraction. In: Joicrtial of Systems arid software, 13(9)(1990)

[8] Kaur Saravjeet and Rashmeet Singh Bali: SQL generation and execution from natural language processing. In: International Journal of Computing and Business Research ISSN (online), pp: 2229-6166(2012)

[9] V. Ambriola,V.Gervasi: Cico: A Tool for Natural Language Requirement Processing. In: Technical report, Dipartimento di In-formatica, Pisa, Italy(1997)

[10] V. Ambriola,V. Gervasi: An Environment for Cooperative Construction of Natural-Language Requirement Bases. In: Proceedings of the Eighth Conference on Software Engineering Environments. IEEE Computer Society Press (1997)

[11] D.M.Berry: The Importance of Ignorance in Requirements Engineering. In: Journal of Systems and Software,28(2): 179- 184 Feb(1995)

[12] James Allen: Natural Language Understanding, First Impression (2007)

[10] Gauri Rao, Snehal Chaudhry and Nikitha Kulkarni, Dr. S H Patil: Natural Language processing using semantic grammar. In: International journal on Computer Science and Engineering, 2(2):219-223.

[14] Marcus.M: A Theory of Syntactic Recognition for Natural Language(1980)

[15] Tomita.M: Efficient Parsing For Natural Language

[16] G. Adomi, M. Zock,: Trends in Natural Language Generation: An Artificial Intelligence Perspective. Number 1036. In: LNCS. Springer, 1993.

[17] Austin,. J.L: How to Do Things with Words(1962)