International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 2 Issue 9, September - 2013

# Syntatic and Semantic Visual Query Building Using Natural Language Processing

Sneha Baskaran
*Meenakshi Sundararajan Engineering College, Chennai*
*Affiliated to Anna University*

Swetha Baskaran
*Meenakshi Sundararajan Engineering College, Chennai*
*Affiliated to Anna University*

Shwetha Gopalarathinam
*Meenakshi Sundararajan Engineering College, Chennai*
*Affiliated to Anna University*

## ABSTRACT

*With the latest advancement in technology and growing data there is a need to extract information in a more efficient and quicker manner using queries. This gives rise to the need for a more easy-to-use query interface. So far, the typical query interfaces are GUI based visual query interfaces. Visual query interfaces however, have limitations especially when they are used for accessing large and complex datasets. The ease of expressing one's queries is limited due to language barrier and the knowledge of precise key words. Therefore, we are developing a novel query interface where users can use natural language expressions to help author visual queries and address the knowledge gap.*

## INTRODUCTION

Decision making is an inherent part across all verticals of businesses. These decisions have to be quick and accurate based on hereditary data retrieved from databases. White collar employees are not technically proficient to handle databases. So working with database query languages like SQL is out of question. To facilitate data access for business owners it has been proposed that natural language interfaces can be used to access to databases. Natural language however has its own disadvantages. Natural language expressions are imperfect and time consuming and require esoteric knowledge to interpret them.

An alternate solution is visual query interfaces. Here the requirements of the user are translated to GUI entities.

Visual query interfaces have their own drawbacks. They are not flexible and often the users need to correlate their requirements to the structure of the database schema. A third party user has superficial knowledge and falls short in expressing his requirements in visual queries.

The aim here is to get an optimized outcome using visual queries in spite of its drawbacks. To do this we propose using natural language expressions to enhance a visual query interface.

## METHODOLOGIES INVOLVED

### 1. SYNTACTIC ANALYSIS

(i) Before getting into actual parsing, we first need to **grade the text input**. Higher grades are given to "main structures" that are often used in input sentences.

(ii) Next comes the actual syntactic parsing. Since syntactic parsing involves the **parsing of multiple sentences**, this step takes a lot of time. In complete parsing, whole words and many of "main structures" in natural language are considered. This parsing is difficult to implement due to the problems that are encountered when information is retrieved. The parser with the identified main structures tries to characterize the possible role for each single word of input text.

One of the most common ways to use all information existing in parse 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),…) as shown in Figure 1.

- **sentence -> noun_phrase, verb_phrase**
- **noun_phrase -> common_noun**
- **noun_phrase -> determiner, distributives**
- **verb_phrase -> verb,**

*Figure 1: Basic structure of a sentences and the components*

Consider the example:

*"Show all the schools"*

This can be parsed as shown in figure 2

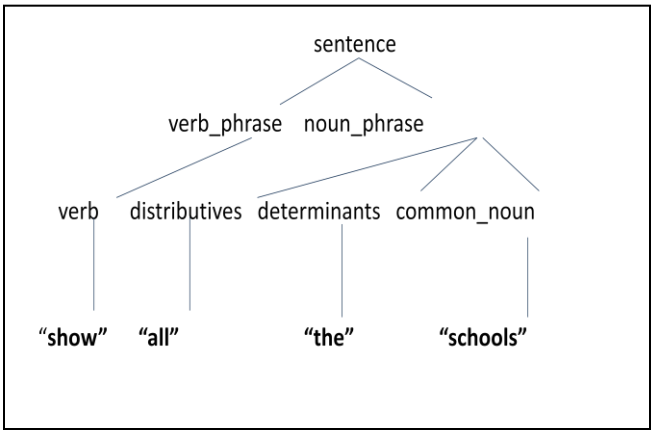


*Figure 2: Syntactic Parsing*

(iii) For implementing an appropriate parsing, we must consider the **lexical cohesion**. It is possible for multiple word expressions to have a different meaning when compared to that of the individual words. These types of expressions are called lexical cohesions. For example the expression "water-hose" has a different meaning when compared to "water" and "hose" considered individually.

## 2. SEMANTIC ANALYSIS

(i) The queries can be expressed in different forms using natural language. For example if a person wants to find a place he could express his query in following ways:

*Which is the north-bound school near Marina Beach?*
*Which is the school in the north of Chennai and near the Marina Beach?*

- We first need to identify all of the entities and their attributes
- Create a list of the synonymous and similar words

For example the attribute "north" in the database can the mapped to synonymous words such as "northern, above, north-bound, top, northward,..."
So with these semantic sets, each of the sentences creates similar SQL query, because they use synonymous words, in other words they there are in same semantic set.

(ii) Next we find the relationship between the words and group them under a single semantic set based on a default attribute e.g. north.

For denoting the relationship between entities and characterizing generic pattern for mapping between user's input and SQL queries, we use particular patterns. Three patterns are shown bas follow:
*<attribute> of <object>*
*<attribute> of <object> of <object>*
*<action verb> object <attribute value>*

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>*

For example if the natural expression given is:
*Which is the school in the north of Chennai and near the Marina Beach?*

The following SQL query will engender:

*SELECT school*
*FROM table*
*WHERE region = "north" and subregion = "Marina Beach"*

## IMPLEMENTION OF OUR PROPOSED SYSTEM

To design automatic system for visual query building, a rule base knowledgebase and a backward chaining inference mechanism is 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.

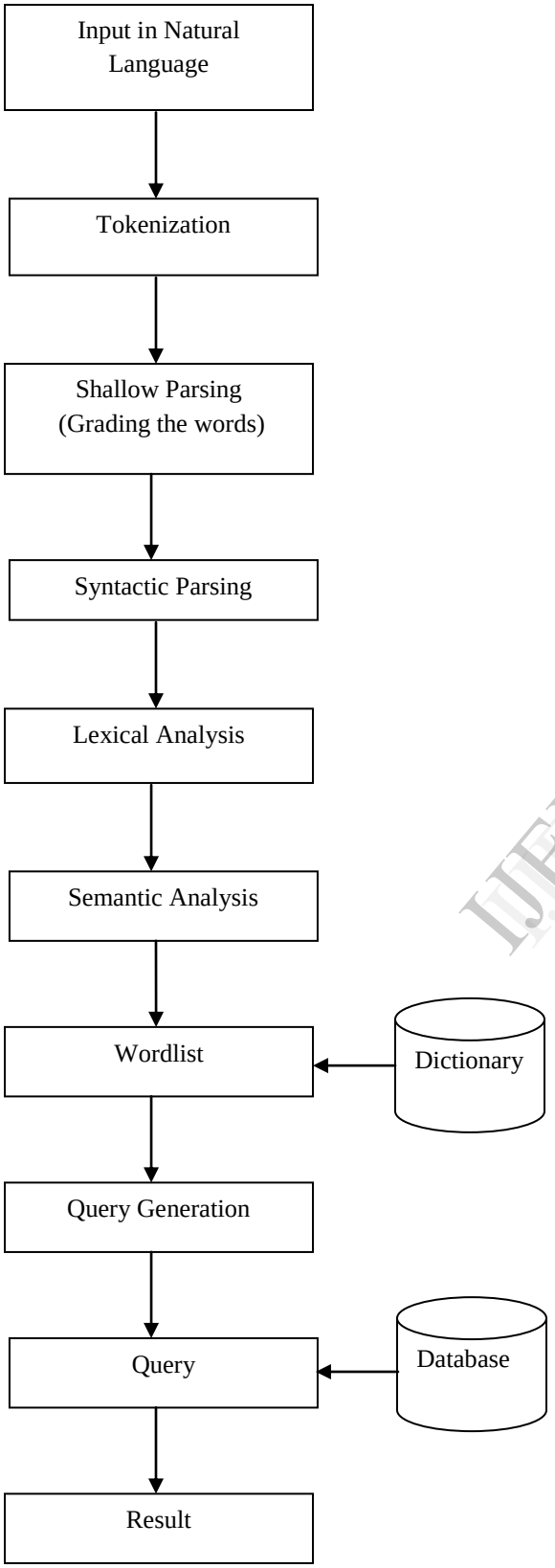


*Figure 3: Flowchart of the methodologies used*

## ALGORITHM

☐ Tokenization (scanning)
- ➢ Split the Query in tokens
- ➢ Give order number to each token identified

☐ Split Query and extract patterns
- ➢ Look for sentence connectors/criteria words
- ➢ Break Query on the basis of connector/criteria tokens.
- ➢ Use criteria tokens to specify condition in query.
- ➢ Find attributes and values after criteria token

☐ Map value for identified attribute and corresponding table

☐ Replace synonyms with proper attribute names

☐ Get intermediate form of Query

☐ Transform it into SQL

## EXAMPLES

Consider a person wants to find a school in north Chennai and near the Marina Beach. To express her data needs, she first creates a school node. She then adds a constraint ("region=north"). As she proceeds to add her second city constraint, "near the Marina Beach", she does not know how to express it. So she leaves the attribute field empty and enters an NL expression "near Marina" in the value field of the constraint combo box. She then submits the query. Given this input, the NL query builder attempts to create a city constraint.

Since the word "Marina Beach" is ambiguous, using the query builder aided by natural language and disambiguate it. After the disambiguation, the query builder completes the GUI Combo box expressing the city constraint: "Subregion = Marina Beach" In this case, the person was able to use NL to author a visual query element (the Subregion constraint) that she did not know how to express initially.

The attribute in the database is "Marina Beach" which the user is not aware of. The visual query builder hence gives her the flexibility of giving the constraint in natural language; for example the user can type in "Marina, Marina Beach, Marina Beach road…" All these are mapped to the attribute "Marina Beach" available in the database.

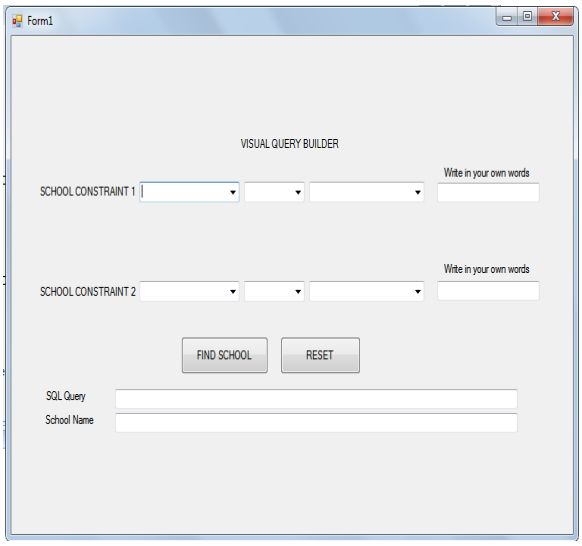


*Figure 4 (i) :The Visual Query Builder*

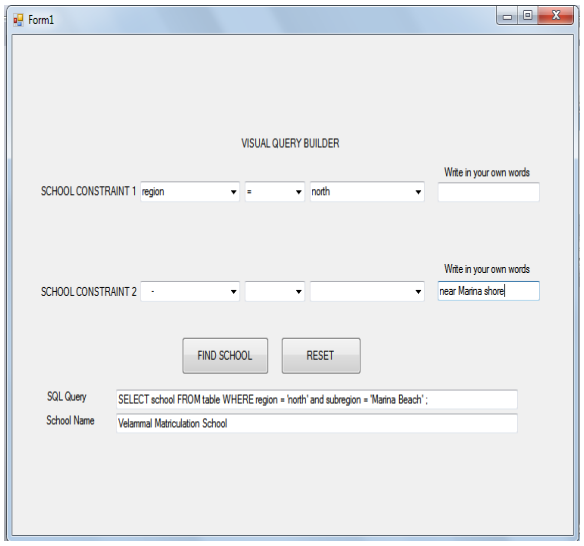


*Figure 4 (iii) : Expressing the constraint 2 in natural language*

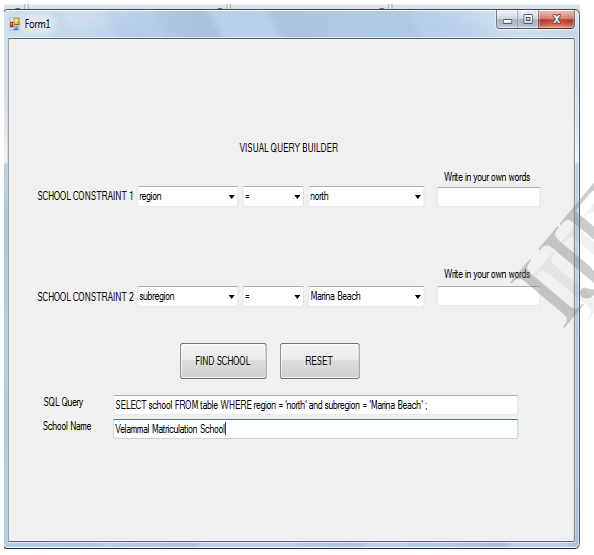


*Figure 4 (ii) : Selecting the constraints*

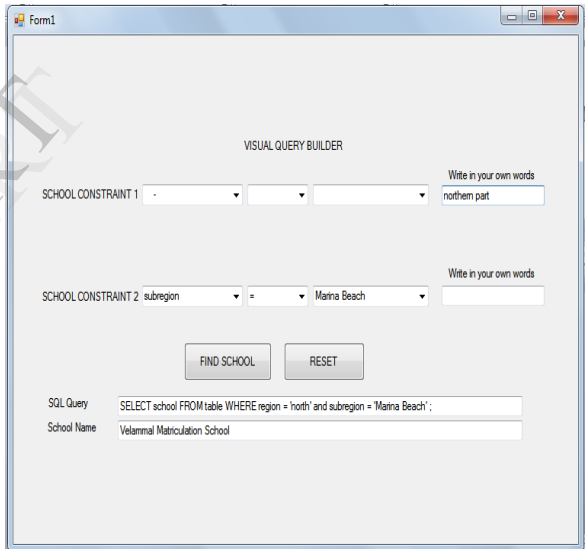


*Figure 4 (iv) : Expressing the constraint 1 in natural language*



## FUTURE ENHANCEMENT

More new grammar can be added to the parser to increase effectiveness. Adding a thesaurus is another suggestion, which could help automating the related words for table and column names.

With the help of a thesaurus, the user input can be pre- processed to substitute related words with table or column names and also remove unwanted words. So far, this system considers *selection* and a few simple aggregations. The next step of the research is, to accommodate more complex queries.

## 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 the terms of input sentence based on searching this knowledge base. This project is presenting the result of using an expert system beside common existing solutions for transforming natural language expressions to SQL query language.

## REFERENCES

[1] Amisha Shingala, Dr. Paresh Virparia
" Enhancing the Relevance of Information Retrieval by Querying the Database in Natural form", *International Conference on Intelligent Systems and Signal Processing (ISSP*),2013
[2] Xuan Xuan, Liu Jianbo1, Yang Jin2
" Research on the Natural Language Querying for remote sensing databases", *International Conference on Computer Science and Service System*,2012
[3]Bouhalika Chouaib, Boufaida Zizette
" Syntactico-Semantic Interpretation of Natural Language Queries on a Medical Ontology", *IEEE Second International Workshop on Advanced Information Systems for Enterprises*,2012
[4] Hasan M Jamil, "Toward a Cooperative Natural Language Query Interface for Biological Databases", *IEEE International Conference on Bioinformatics and Biomedicine*,2011
[5]Alex Berson,Stephen J.Smith"Data warehousingData Mining,&OLAP",*Tata McGraw -Hill Edition*, 2004