I.J. Intelligent Systems and Applications, **2013, 12, 81-90**
Published Online November 2013 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijisa.2013.12.07

Semantic Analysis of Natural Language Queries Using Domain Ontology for Information Access from Database

Avinash J. Agrawal
Ramdeobaba College of Engineering & Management, Nagpur, India
E-mail: avinashjagrawal@gmail.com

Dr. **O. G. Kakde**
Veermata Jijabai Technological Institute, Mumbai, India
E-mail: directorvjti@vjti.org.in

***Abstract*—**This paper describes a method for semantic analysis of natural language queries for Natural Language Interface to Database (NLIDB) using domain ontology. Implementation of NLIDB for serious applications like railway inquiry, airway inquiry, corporate or government call centers requires higher precision. This can be achieved by increasing role of language knowledge and domain knowledge at semantic level. Also design of semantic analyzer should be such that it can easily be ported for other domains as well. In this paper a design of semantic analyzer for railway inquiry domain is reported. Intermediate result of the system is evaluated for a corpus of natural language queries collected from casual users who were not involved in the systemdesign.

***Index Terms*—** Natural Language Interface to Database, Semantic Analysis, Domain Ontology, Language Modeling

I. Introduction

In natural language interface to database (Androutsopoulos et al., 1995), the information seeker uses natural language for submitting his or her query instead of structured database query. Natural language interface to database is not a new area. A lot of research is going on since a long time (Allen, 1994). Natural languages can act as an alternative interface for getting structured information from database. Writing questions in natural language is much easier for a casual user than the traditional graphical user interface for database access. The graphical user interface which is mostly used for database access is complicated and requires time consuming navigation and this becomes even worse in small devices. NLIDB shifts a user's burden of learning use of interface to describe his or her need for information to the system. NLIDB thus demands less input-output and processing facilities which make it more useful particularly for small handheld devices like mobile phone.

Most of the existing NLIDB systems are designed using shallow semantic analysis of inputted natural language query (Popescu et al., 2004). It results in low success rate and too much dependency on database schema due to which practical implementation of such system is not feasible especially for serious usage of database information. Implementation of NLIDB for applications like inquiry in railways, bank, corporate, government organization etc. requires detail analysis of input query for interpretation of its intended meaning. Detail analysis helps in increasing success rate. After detail analysis the intended meaning can be represented in some intermediate form which reduces database dependency.

For detail analysis of input query use of language knowledge and domain knowledge is to be increased at semantic level (Minock, 2005). Here a method is proposed to use ontology to represent domain knowledge and language modeling to represent language knowledge. For experimentation of the proposed method, domain ontology for railway inquiry is populated. A Semantic analyzer is designed using language modeling of English queries which analyzes the preprocessed English language query to interpret its intended meaning. After interpretation the meaning is represented in domain ontological form. Using this intermediate form of meaning corresponding database query can be generated for the target database.

For evaluation, the system is tested on a corpus of English language query which is collected from various groups of user of the railway inquiry domain. These users are essentially not aware of the internal system design.

Next section discusses related work in the area, then section III describes the concept of domain ontology, section IV proposes system architecture design. In section V implementation of the system is explained in detail. The implemented system is then evaluated and result is algorithm is widely adopted in real engineering given in section VI. Finally section VII contains conclusion and future scope.

II. Related Work

Natural language interface to database is not a new topic, research is going on since a long time many systems are suggested and still it is an open problem. The first QA systems were developed in 1960s and they were basically NL interfaces to expert systems, tailored to specific domains, the most famous ones being BASEBALL and LUNAR. Both systems were domain specific, the former answered questions about the US baseball league over the period of one year, the later answered questions about the geological analysis of rocks returned by the Apollo missions. LUNAR was able to answer 90% of the questions in its domain when posed by untrained geologists. Some of the early NLIDBs approaches uses simple pattern-matching techniques. In the example described by (Androutsopoulos et al., 1995), a rule says that if a user's request contains the word "capital" followed by a country name, the system should print the capital which corresponds to the country name, so the same rule will handle "what is the capital of Italy?", "print the capital of Italy", "could you please tell me the capital of Italy". This shallowness of the pattern-matching would often lead to failures but it has also been an unexpectedly effective technique for exploiting domain-specific data sources. The main drawback of these early NLIDBs systems is that they were built having a particular database in mind, thus they could not be easily modified to be used with different databases and were difficult to port to different application domains. Configuration phases were tedious and required a long time, because of different grammars, hard-wired knowledge or hand-written mapping rules that had to be developed by domain experts.

The next generation of NLIDBs used an intermediate representation language, which expressed the meaning of the user's question in terms of high-level concepts, independently of the database's structure (Androutsopoulos et al., 1995), making a separation of the (domain-independent) linguistic process and the (domain-dependent) mapping process into the database to improve the front end portability (Martin et al., 1985). The formal semantics approach presented in (De Roeck et al., 1991) made a clear separation between the NL front ends, which have a very high degree of portability, and the back end. The front end provides a mapping between sentences of English and expressions of a formal semantic theory, and the back end maps these into expressions, which are meaningful with respect to the domain in question. Adapting a developed system to a new application involves altering the domain specific back end alone.

MASQUE/SQL (Androutsopoulos et al., 1993) is a portable NL front end to SQL databases. It first translates the NL query into an intermediate logic representation, and then translates the logic query into SQL. The semi-automatic configuration procedure uses a built-in domain editor, which helps the user to describe the entity types to which the database refers, using an is-a hierarchy, and then to declare the words expected to appear in the NL questions and to define their meaning in terms of a logic predicate that is linked to a database table/view.

More recent work in the area can be found in PRECISE (Popescu, et al, 2004). PRECISE maps questions to the corresponding SQL query by identifying classes of questions that are understood in a well defined sense: the paper defines a formal notion of *semantically tractable* questions. Questions are sets of attribute/value pairs and a relation token corresponds to either an attribute token or a value token. Each attribute in the database is associated with a wh-value (what, where, etc.). Also, a lexicon is used to find synonyms. The database elements selected by the matcher are assembled into a SQL query, if more than one possible query is found, the user is asked to choose between the possible interpretations. However, in PRECISE the problem of finding a mapping from the tokenization to the database requires all tokens to be must distinct; questions with unknown words are not semantically tractable and cannot be handled. As a consequence, PRECISE will not answer a question that contains words absent from its lexicon. Using the example suggested in (Popescu, et al, 2004), the question "what are some of the neighbourhoods of Chicago?" cannot be handled by PRECISE because the word "neighbourhood" is unknown. When tested on several hundred questions, 80% of them were semantically tractable questions, which PRECISE answered correctly, and the other 20% were not handled.

NLIDB have attracted considerable interest in the Health Care area. In the approach presented in (Hallet et al., 2007) users can pose complex NL queries to a large medical repository, question formulation is facilitated by means of Conceptual Authoring. A logical representation is constructed using a query editing NL interface, where, instead of typing in text, all editing operations are defined directly on an underlying logical representation governed by a predefined ontology ensuring that no problem of interpretation arises.

However, all these approaches still need an intensive configuration procedure. To reduce the formal complexity of creating underlying grammars for different domains, (Minock et al., 2008), and most recently C-PHRASE (Minock et al., 2010) present a state-of-the-art authoring system for NLIDB. The author builds the semantic grammar through a series of naming, tailoring and defining operations within a web-based GUI, as such the NLI can be configured by non-specialized, web based technical teams. In that system queries are represented as expressions in an extended version of Codd’s Tuple Calculus and context-free synchronous grammars extended with lambda functions to represent semantic grammars, which may be directly mapped to SQL queries or first-order logic expressions. High-order predicates are also used to support ranking and superlatives.

Above mentioned work suggest that after initial work in 1960’s when NLP was in its infancy again since 1990’s work in this area is continuously progressing. This progress is moving with progress in the area of NLP. As now use of NLP is started in many application areas, work in the field of NLIDB should also aim at its usage in real life application. For this researchers have to work for increasing success rate, portability and robustness of application specific systems.

III. Domain Ontology

For semantic analysis some knowledge representing structures is used. There are various methods to represent knowledge. Selection of knowledge representation method depends on the application domain and the task at hand. Method to represents knowledge also decides the kind of semantic analysis can be done i.e. its complexity, portability, robustness and comprehensiveness. Broadly semantic analysis is of two types shallow and deep. Shallow semantic analysis is a method of superficial analysis, while deep semantic analysis is a method of detail analysis. Shallow semantic analysis is used to reduce the complexity of the development. Shallow analysis uses domain knowledge and keyword searching to interpret meaning which results in low success rate. For natural language interface to database deep semantic analysis increases success rate and portability compared to its counterpart shallow method. Deep semantic method uses previously defined semantic structures like First order predicate logic, Phrase Structured Grammar, frame structure, semantic network, ontology etc. to represent meaning (Venessa et. al. 2011). Here domain ontology is selected as a semantic structure for the domain under consideration (Railway Inquiry) which is described below.

Ontology (Jurafsky, 2008) is a model of the world, represented as a tangled tree of linked concepts. Concepts are language-independent abstract entities, not words. The purpose of the Ontological Semantic is to improve automated text processing by providing language independent, meaning based representations of concepts in the world. The ontology shows how concepts are related and what properties each have. Unlike words in a language, each ontological concept is unambiguous.

There are three major types of concepts in ontology,

•**OBJECTs**: the static things that exist in the physical (e.g., TRAIN), mental (e.g., CONCESSION) and social (e.g., DEPARTMENT) world.

•**EVENTs**: any activities, actions, happenings or situations: i.e., things that occur over some period of time. This includes physical events, like RUN, mental events, like late.

•**PROPERTYs**: describes properties of OBJECTs and EVENTs. Properties are used to define other concepts in the ontology, in most cases by linking related concepts: e.g., EXPRESS is related to TRAIN by the concept CATAGORY-OF (express is the category of train).

Objects and events are identified for the railways domain and also analysis of how they are related to each other is done. Properties to describe each concept of the considered domain are found. The detailed ontology structure given in Fig.1 is designed. To develop this ontology structure, information related to domain is collected from various sources such as railway guide, magazines and websites. The corpus of questions collected after surveying various users is also used to develop this ontology structure.

The ontology structure of Fig.1 contains major concepts involved in railway domain like Train, Station, Seats, Fare and Concession etc. These concepts are described with the associated property set. For example properties train number, train name, type, status etc. are describing the concept train. Some property links one concept with other and thus defines relations between concepts. For example concept Train is related to concept Station by From (Source Station for a Train) property. Thus the ontology designed represents the world of railways inquiry.

IV. Proposed Work

Semantic analysis is very important part of any natural language processing system. It determines the meaning of given sentence and represents that meaning in an appropriate form. Basically there are two types of semantic analysis shallow and deep (Wong, 2005). Shallow semantics is superficial, faster but less accurate. Deep method uses detailed language knowledge which is less flexible but more accurate. Here a deep method is proposed in which the input is analyzed at word level and sentence level using language knowledge. Ontological representation of semantic is used, where meaning is represented in the form of a graph.

In natural language interface to database, intension is to determine the meaning of input question from the perspective of database concepts as finally a database query needs to be generated from the question which will actually brings the desired information from the database.

In a database query there are two important constituents (Leonardo, 2008):

- What is expected and
- What are constraints

Every database query contains these two constituents. So if database query equivalent to the natural language question is to be generated then the natural language question is to be analyzed to locate the part of the question which contributes to this information. In semantic analysis of natural language question these two constituent needs to be identified.

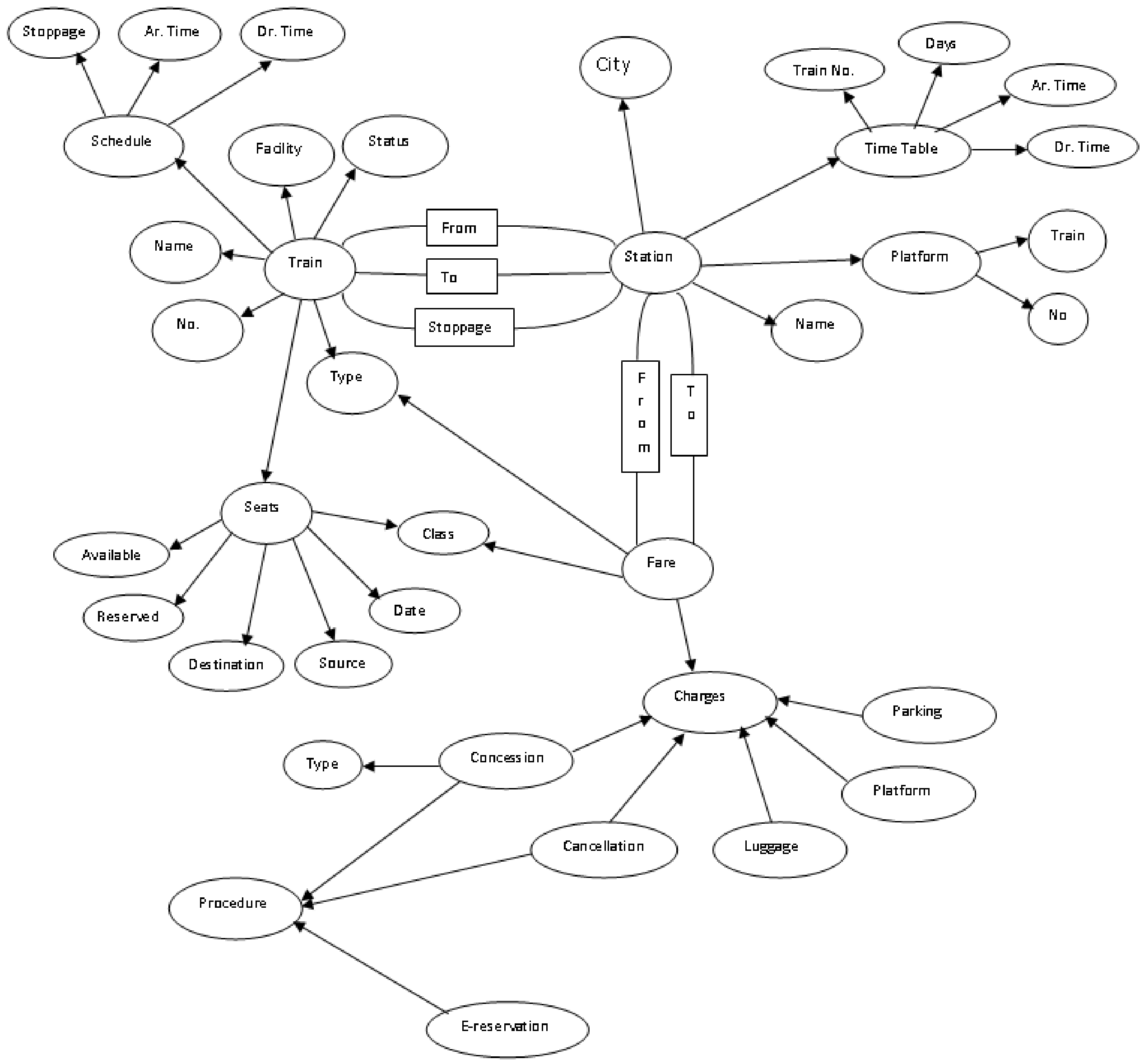

Fig. 1: Domain ontology structure for Railway Inquiry

Before semantic analysis the natural language question is to undergo preprocessing. This preprocessing involves pos tagging, chunking and named entity recognition. Part of speech tagger tags all words of natural language question with its pos category. Chunking groups related words into meaningful sentence constituents. Named entity recognizer identifies named entities in the natural language question and classifies it into semantic category. It relates named entity with appropriate properties of concepts in domain ontology.

A typical output of preprocessing is:

```
                                   [Train(Name)]
 What  is   the  position  of   gitanjali   express
| W   |Aux| Det|   Noun   |Prep|  PNoun |   Noun  |
| W   |Aux|  Noun Phrase  |Prep|   Noun Phrase    |
| W   |Aux|Noun Phrase    |      Prep Phrase      |
| W   |              Verb Phrase                  |
|                   Sentence                      |
```

It is observed that named entities identified and classified in preprocessing of the input query are always used as constraints. In above example named entity “gitanjali” adds a constraint “Train(Name) = gitnajali” in the constraint list of the natural language question. A natural language question may have any number of named entity results in same number of constraints in the corresponding constraint list. Named entity is not the only source of constraints component of corresponding database query. There are other sources also which is explained in next paragraphs.

To identify the expected entity language modeling of questions is required. Using language modeling the important words which contribute in deciding the expected entity are identified. For example, in questions starting with the word ‘what’ the noun phrase immediately after the wh-word determines the expected entity. Similarly if question starts with ‘when’ the verb of sentence determines the expected entity. In the question ”What is the position of gitanjali express” NP after wh word is “the position” determines what is expected. Similarly in query “When howrah mail reaches nashik” verb of sentence ‘reaches’ determines the expected entity. A rule base is generated using such rules for a wide variety of questions. It includes questions which starts from wh-words, list type questions, questions which starts from aux-verbs like do, is etc. This rule base is independent of the domain under consideration hence it can be reused for other domain as well. The purpose of the rule base is to extract the important words which contribute to decide expected entity in the question. These important words are then needs to map with standard concepts of ontology.

To determine constraints, it is observed in the corpus under consideration that many times named entities are sufficient. But sometimes non named entity words also contribute to add constraints. In some questions the words which determines the expected entities also adds constraints like adjectives and adverbs. For example in question “List superfast trains for Nagpur Mumbai” the expected entity as per the language modeling is “superfast trains” contains adjective “superfast” results in addition of constraint: “ train type is superfast”. Such words which contribute in generating additional constraints are also identified from the expected entity which is captured after language modeling of the natural language question. A typical output of language modeling is:

For example query:

“list superfast trains from Mumbai to Delhi”

Expected entity is: *superfast train*

Contraints :

1) *Train[from] = Mumbai*{constraint from named entity}

2) *Train[to] = Delhi* {constraint from named entity}

3) *Train[type] = Superfast* {additional constraint}

These values i.e. expected entity and constraints needs to be mapped to the standard concepts and properties in the domain ontology. After mapping, the intermediate representation of the natural language question will be as shown in Fig.2. This graph which is a sub-graph of the larger ontology graph of Figure 1 represents the meaning of the given natural language question. This graph is expected output of the semantic analysis phase.

For mapping expected entity with standard concepts of ontology a lexicon is used. The lexicon is domain dependent and developed for the domain under consideration. This lexicon can also be considered as an extension of domain ontology. The block diagram of the method is given in Fig.3.

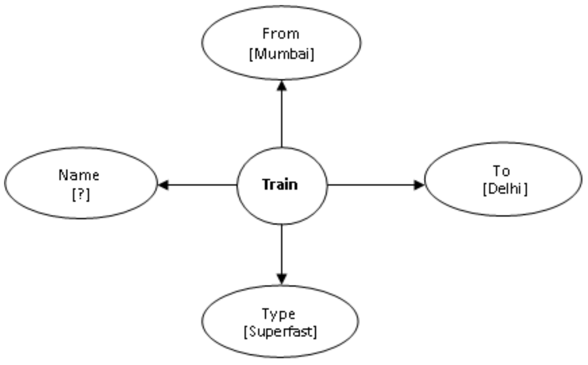

Fig. 2: Sample output of semantic analyzer

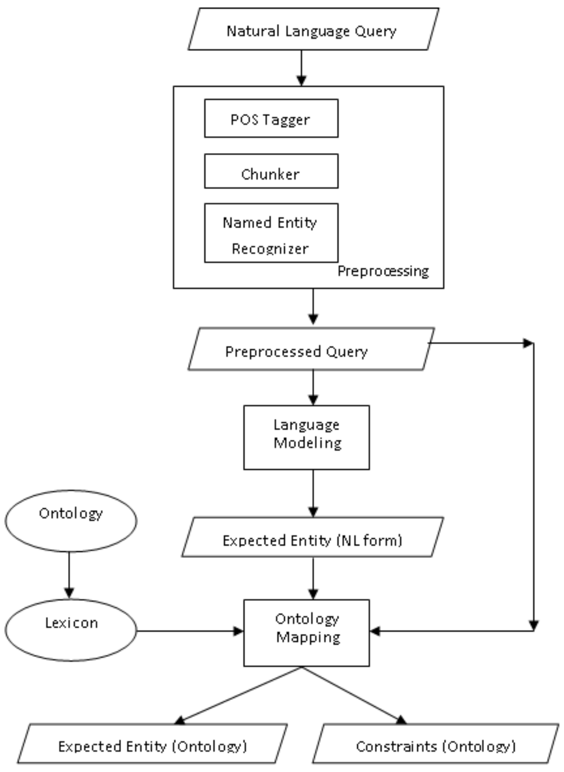

Fig. 3: Block diagram for Semantic Analysis

V. Implementation

To test the proposed work, a test corpus i.e. a set of natural language queries for the domain under consideration (i.e. Railway inquiry) is prepared after exhaustive survey on actual users of railway who were not involved in the system design. These queries are first given to the preprocessing module. The preprocessing module has three steps: pos tagging, Chunking and named entity recognizer. Natural language Toolkit with python is used for the preprocessing. POS tagger tags each word of query by corresponding part of speech category. Then the queries are given to the Parser which analyzes the syntax of given natural language query and finds relation between words. A context free grammar of all possible natural language queries is designed. Parser uses this context free grammar to generate parse tree of the given natural language queries. Output of parser is a parse tree contains various constituents of the input question like noun phrase, verb phrase, adjectival phrase etc. This output is then given to named entity classifier which will classify all named entity used in natural language query into its semantic category. This part of preprocessing is domain dependent. Here supervised machine learning method is to classify named entity (Shende, 2012). Using a training corpus n-gram is generated which is then used to classify named entity.

After this preprocessing the preprocessed query is then submitted to semantic analyzer which uses language modeling to extract expected entity and constraints. In the language modeling used, starting words are very important. Based on starting words of natural language question the part of it which contribute the most in deciding expected entity is to be located. Rules are written to extract these important words from the input query. These rules use knowledge of English language and are completely domain independent. Same set of rules can be used for other domain as well. The part of natural language query extracted by language modeling mapped with terms of ontology. For this mapping a lexicon is used which uniquely maps expected entity of natural language to entities and corresponding properties of domain ontology. To search the expected entity of natural language in lexicon best search method is used. The nearest match of the expected entity is found in lexicon (Tablan, 2008). The corresponding entry of lexicon contains related ontology entities, properties and some additional constraints. These additional constraints are added to the list of constraints due to occurrence of named entity. The ontology entity and property found in lexicon entry is finally stored as output of semantic analyzer along with the list of constraints in an intermediate file for further processing i.e. generation of corresponding database query.

A. Ontology Mapping

For mapping expected entity extracted from language modeling and constraints from named entity recognizer with ontology terms semantic lexicon is used which is an extension of ontology. A lexicon is a list of words in a language, a vocabulary along with some knowledge of how each word is used. A lexicon may be general or domain-specific, for example, a lexicon of several thousand common words of English or some language. The words that are of interest are usually open-class or content words, such as nouns, verbs, and adjectives, rather than closed-class or grammatical function words, such as articles, pronouns, and prepositions, whose behavior is more tightly bound to the grammar of the language. A lexicon may also include multi-word expressions such as fixed phrases (by and large), phrasal verbs (tear apart), and other common expressions.

The lexicon has a highly semantic structure that governs what words can mean, and how they can be used. This structure consists of relations among words and their meanings, as well as the internal structure of individual words. The linguistic study of this systematic, meaning related, structure is called lexical semantics. In the proposed system syntactic phrases or some sequence of words are used as an individual entry in the lexicon. Thus lexicon is finite set of these phrases or word sequence. Each entry of lexicon is paired with its corresponding terms in ontology. Ontology term include related entity and properties. Also it contains additional constraints (other than named entity) if required. This lexicon is able to map expected entity extracted from language modeling to standard terms of ontology. This mapping cannot be one to one as it is not necessary in natural language to use same words to refer a concept. In natural language one thing may be expressed in many ways which include choice of words, word sequence etc. Moreover language writing mistakes such as spelling errors, syntax errors etc. are also needs to be absorbed. Due to this instead of strict one to one mapping nearest map is normally used. In the proposed system similarity matching method is used which is given in next section.

B. Similarity Matching

Measures of semantic similarity between concepts are widely used in Natural Language Processing. Similarity evaluation between two documents is an important operation which lies at the heart of most text and language processing tasks. The similarity evaluation forms a main part of the information retrieval system for retrieving the information. Some parameters need to be similar so that the related thing from the large bunch of materials can be retrieved. The similarity matching are used at many places like, use of a search engine where web-page documents are requested which bear some similarity to the keywords or string(s) which constitute the query document. Similarity matching means matching the more similar word to the specified word. Based on the entity, properties the matching plays an important role in identifying the more appropriate word to match with them.

Considering the above examples related to the use of similarity matching one is able to compute the partial goal of the system like classification, validation, generation, etc. A measure of semantic similarity takes as input two words or phrases, and returns a numeric score that quantifies how much they are alike. Such a measure is usually based on is –a relations found in the underlying taxonomy or ontology in which the words or phrases reside. For example, express train and train are similar in which express train is a kind of train. Likewise, express train and fast train are similar in that they are both kinds of trains. Of course, many ontology include additional relations between concepts such as has-part, is-a-way-of-doing, is-belongs-to etc that are not directly accounted for measures of similarity.

Thus, semantic similarity is viewed as a special case of semantic relatedness, and it is believed that developing measures that take advantage of increasingly rich ontology (particularly in the domain under consideration) which has a wealth of relations is an important area of research.

Here in this system semantic similarity is used due to the common features of natural languages. Some of the features are: Natural language sentences are incomplete descriptions of the information that they are intended to convey. The same sentence means different things in different contexts. No natural language can be complete because new words, expressions, and meanings can be generated quite freely. There are lots of ways to say the same thing. While searching lexicon the system is not able to match the exact similarity between the pos tagged words and the entries in semantic lexicon because of above mentioned problems of natural languages. In addition to this there may be various errors in the input query like spelling mistakes, grammatical errors etc. Such errors in input query are to be considered while evaluating similarity. Avoiding the human language error problem can become a great challenge for the system as the input is provided by untrained users.

To address this problem some similarity matching method is to be implemented that matches the important words extracted from the input query with lexicon entries. This method may give some value to indicate similarity between two entities. The lexicon entry having highest value of similarity function for the input natural language query will be considered. Thus nearest matching is used to get lexicon entry for mapping natural language terminology with ontology. There are various methods available for similarity matching. In this thesis Jaccard similarity method is implemented.

Jaccard similarity determines the Jaccard coefficient. The Jaccard similarity coefficient is a statistical measure of similarity between sample sets. For two sets, it is defined as the cardinality of their intersection divided by the cardinality of their union.

Mathematically,

$$J(A, B) = |A \cap B| / |A \cup B|$$

For a pair of entities, the calculation is performed on sets of entities with which the elements of the pair occur. For example, in calculating the similarity of pairs of pages, the numerator is the number of users that have edited both pages, and the denominator is the number of users who have edited either or both.

Mathematically,

$$J(A, B) = |X \cap Y| / |X \cup Y|$$

Where, X and Y correspond to the sets of entities that occur with A and B, respectively.

Eg: $X = \{A, B, C, D, E\}, Y = \{B, C, D, E, F\}$

Jaccard similarity = 4/6 = 0.67

In this way the Jaccard similarity method works and calculates the Jaccard coefficient. This value is then used to determine similarity between two entities. In some usage a threshold value is set to declare similarity if coefficient value exceeds some predefined value. In other usage highest value of this coefficient is selected to declare nearest match.

In this system similarity matching is used to map linguistic phrases with ontology terms such as entities and properties. From the training corpus lexicon is populated. Each natural language query in training corpus is semantically analyzed after preprocessing to get linguistic phrase which contribute in deciding expected entity. These linguistic phrases are then entered in to lexicon along with respective ontology entity and property. While testing such linguistic phrase is extracted through semantic analysis in the same way. For this phrase corresponding entry in lexicon is to be found. It is not necessary that this linguistic phrase is present in the lexicon. Here similarity matching is used to find nearest match to get corresponding entity and property in ontology. Jaccard coefficient is calculated for the input natural language query and all entries in lexicon. The entry of lexicon having highest value of Jaccord coefficient is considered and corresponding entity and property is retrieved.

A typical output of language modeling is:

For example query:

"list superfast trains from Mumbai to Delhi"

Expected entity is: *superfast trains*

Contraints :

1) *Train[from] = Mumbai*{constraint from named entity}

2) *Train[to] = Delhi* {constraint from named entity}

Now this value of expected entity "superfast trains" needs to be mapped with standard concepts and properties in the domain ontology. Jaccord coefficient is calculated for the linguistic phrase "superfast trains" and entries in lexicon. Using this similarity matching, lexicon entry for "superfast train" is selected due to highest value of Jaccord coefficient. This entry contains train as entity and train name as its property and in addition it contains a constraint as train type is superfast. So after mapping, the query will be represented as:

Expected entity (ontology): *train*

Expected property (ontology): *name*

Contraints :

1) *Train[from] =Mumbai*{constraint from named entity}

2) *Train[to] = Delhi* {constraint from named entity}

3) *Train[type] = Superfast* {additional constraint}

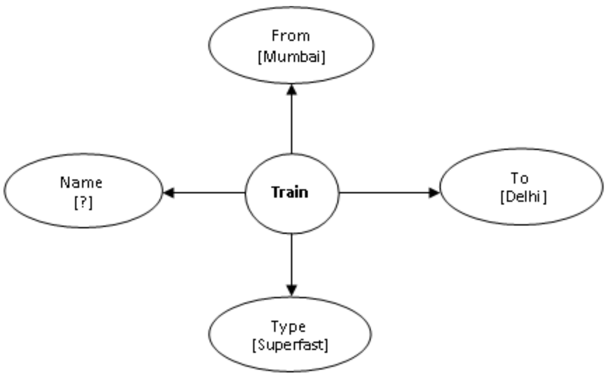

Fig. 4: Graphical representation of Semantic

This intermediate representation of query which is output of semantic analysis may be represented in graphical form as shown in Fig.4. This graph which is a sub-graph of the larger ontology graph of Fig.1 represents the meaning of the given natural language question. Database query needs to be generated from this representation.

VI. Evaluation

To evaluate the methodology correctness of the semantics stored in the intermediate file is to be checked with corresponding natural language query of the testing corpus. This intermediate file is evaluated for correctness in information extraction. Manually all semantics generated in intermediate file is checked and number of correct interpretation, incorrect interpretation and no response is counted. For evaluation of correctness two parameters are used precision and recall given by the following formula:

Precision = correct / (correct + incorrect)

Recall = correct / (correct + incorrect + no response)

High precision means that an algorithm returned more relevant results than irrelevant. High recall means that an algorithm returned most of the relevant results. Thus precision is the measure of how much relevant results are produced compare to irrelevant and the recall is the measure of how many relevant results are produced. Values of the measures are calculated and are shown in the Table 1. Correctness of semantic interpretation is also calculated for major questions and answer types and are given in Table 2 and Table 3. Question types basically depend on the first word of the question and answer type is depends on expected answer of the question. Graph for correctness on question and answer types are also plotted.

Table 1: Experimentation results

Total Number of Input NL query	Correct Interpretation	Incorrect Interpretation	No Response	Precision	Recall
300	271	13	16	95.42	90.33

Table 2: Distribution of correct answer over question types

Question Type	Correctness in % (Precision)
What	92.04
When	91.66
Is	87.50
Does	85.71
List	87.50
How many	90.38
How much	85.18
Whether	87.50
By what	93.33

Table 3: Distribution of Correct answer over answer type

Answer Type	Correctness in % (Precision)
Train Name	96.83
Arrival Time	89.28
Departure Time	100
Current Status	83.34
Seat Availability	89.47
Fare	87.50
Stoppage	83.33
Facilities	62.50
Concession	76.92
Procedure	82.36
Platform	100

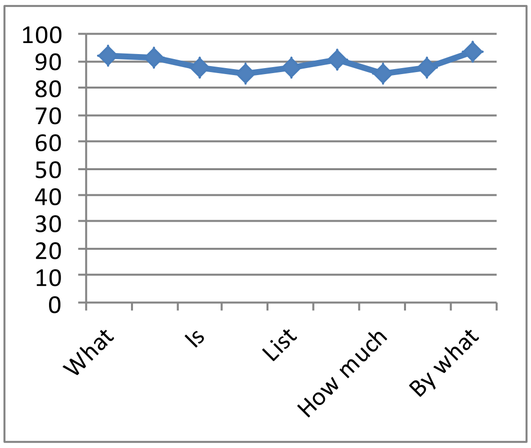

Fig. 5: Distribution of Correctness over question types

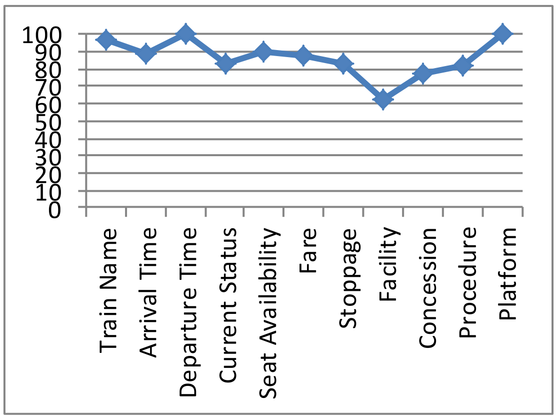

Fig. 6: Distribution of Correctness over Answer types

VII. Conclusion and Future Scope

Natural language interface to database is a technology which can be used to access information from database by a casual user. It can be proved to be very useful for many applications like railway/airway inquiry, corporate inquiry, government information systems, call centers etc. For such serious applications high precision value is required. By increasing use of language knowledge in semantic analysis higher accuracy can be achieved. Another important parameter is portability from database and domain perspective. In the method discussed no knowledge of database is used for semantic interpretation so it is database independent. Domain knowledge is used as separate external resources which make it usable for other domain as well after replacing those resources by concerned domain knowledge.

In future, work is to be done to generate database query from this semantic interpretation for which actual database terminologies are to be mapped with the terms of domain ontology. Also this system is to be ported on handheld devices like mobile phones which will make it more useful general user. Further if speech interface can be incorporated then it can prove to be very useful technique for database interface.

Refrences

[1] Allen James. Natural Language Understanding. A Book Published by Addison Wesley Publications, 1994.

[2] Androutsopoulos, I., Ritchie, G.D., and Thanisch, P. MASQUE/SQL - An Efficient and Portable Nat-ural Language Query Interface for Relational Databases. In Proc. of the 6th International Conference on In-dustrial and Engineering Applications of Artificial Intel-ligence and Expert Systems, Edinburgh, Gordon and Breach Publisher Inc, 1993, p.327-330.

[3] Androutsopoulos I., Ritchie G.D., and Thanisch P. Natural Language Interfaces to Databases – An Introduction. Journal of Natural Language Engineering Part 1, 1995, 29–81.

[4] Ana-Maria Popescu, Alex Armanasu, Oren Etzioni, David Ko, and Alexander Yates. Modern Natural Language Interfaces to Databases: Composing Statistical Parsing with Semantic Tractability. COLING, 2004.

[5] De Roeck, A., N., Fox, C., J., Lowden, B., G., T., Turner, R., Walls, B. A Natural Language System Based on Formal Semantics. In Proc. of the International Con-ference on Current Issues in Computational Linguistics, Penang, Malaysia, 1991.

[6] Hallett, C., Scott, D. and Power, R. Composing Questions through Conceptual Authoring. Journal of Computational Linguistics, MIT Press, 2007, 33(1): 105-133

[7] Jurafsky D., Martin J. Speech and Language Processing. A Book Published by Prentice Hall Publications.2008.

[8] Leonardo Lesmo. Natural Language Query interpretation in restricted domains. In the procedding of 6th International Conference on NLP ICON-08, Pune, India, 2008.

[9] Martin, P., Appelt, D., E., Grosz, B., J., Pereira, F., C., N. TEAM: An Experimental Transportable Natural Language Interface. IEEE Database Engineering. 1985, 8(3): 10-22.

[10] Minock, Michel. Where are the killer application of restricted domain question answering. In proceeding of the IJCAI Workshop on Knowledge Reasoning in Question Answering, page 4, Edinburgh, Scotland, 2005.

[11] Minock Michael, Olofsson Peter, Naslund Alexander. Towards Building Robust Natural Language Interface to Database. In the proceeding

of 13th International Conference on Applications of Natural Language to Information Systems, NLDB, London, UK, 2008, 187-198.

[12] Minock, M. C-Phrase: A System for Building Robust Natural Language Interfaces to Databases. Journal of Data Engineering (DKE), 2010, 69(3): 290-302.

[13] Popescu Ana-Maria, Etzioni Oren, and Kautz Henry. Towards a Theory of Natural Language Interfaces to Databases. IUI, 2003, 149–157.

[14] Shende A., Agrawal A. J., Kakde O. G. Domain Specific Named Entity Recognition Using Supervised Approach. International Journal of Computational Linguistics (IJCL), 2012, Volume (3), Issue (1), page 67-78.

[15] Tablan, V, Damljanovic, D., and Bontcheva, K. A Natural Language Query Interface to Structured Information. In the Proceeding of 5th European Semantic Web Conference, Tenerife, Spain. Springer Verlag, 2008, 1-15.

[16] Vanessa Lopez, Victoria Uren, Marta Sabou, Enrico Motta. Is Question Answering fit for the Semantic Web?: a Survey. Semantic Web Journal, volume 2, number 2, 2011,page 125-155.

[17] Yuk Wah Wong. Learning for Semantic Parsing Using Statistical Machine Translation Techniques, Technical Report UT-AI-05-323, University of Texas at Austin, Artificial Intelligence Lab, 2005.

Authors' Profiles

Avinash J. Agrawal received Bachelor of Engineering Degree in Computer Technology from Nagpur University, India and Master of Technology degree in Computer Technology from National Institute of Technology, Raipur, India in 1998 and 2005 respectively. He is currently pursuing Ph.D. from Visvesvaraya National Institute of Technology, Nagpur. His research area is Natural Language Processing and Databases. He is having 15 years of teaching experience. Presently he is Assistant Professor in Shri Ramdeobaba Kamla Nehru Engineering College, Nagpur. He is the author of several research papers in International and National Journal, Conferences.

Dr. O. G. Kakde received Bachelor of Engineering degree in Electronics and Power Engineering from Visvesvaraya National Institute of Technology (formerly Visvesvaraya Regional College of Engineering), Nagpur, India and Master of Technology degree in Computer Science and Engineering from Indian Institute of Technology, Mumbai, India in 1986 and 1989 respectively. He received Ph.D. from Visvesvaraya National Institute of Technology, Nagpur, India in 2004. His research interest includes theory of computer science, language processor, image processing, and genetic algorithms. He is having over 25 years of teaching and research experience. He worked as Professor and Dean, Research and Development at Visvesvaraya National Institute of Technology, Nagpur, India. Presently he is working as Director at Veermata Jijabai Technical Institute, Mumbai, India. He is the author or co-author of more than thirty scientific publications in international journals, international conferences, and national conferences. He also authored five books on data structures, theory of computer science, and compilers. He is the life member of Institution of Engineers, India. He also worked as the reviewer for international and national journals, international conferences, and national conferences and seminars.

How to cite this paper: Avinash J. Agrawal, O. G. Kakde,"Semantic Analysis of Natural Language Queries Using Domain Ontology for Information Access from Database", International Journal of Intelligent Systems and Applications(IJISA), vol.5, no.12, pp.81-90, 2013. DOI: 10.5815/ijisa.2013.12.07