INTELLIGENT QUERY PROCESSING IN MALAYALAM

RAJI SUKUMAR A[1] AND BABU ANTO P[2]

[1]Department of Information Technology, Kannur University, Kannur , Kerala India.
rajivinod.a@gmail.com
[2] Department of Information Technology, Kannur University, Kannur , Kerala India.
bantop@gmail.com

ABSTRACT

The paper presents a model for developing intelligent query processing in Malayalam. For this the investigator has selected a domain as time enquiry system in Malayalam language. This work discusses issues involved in Natural Language Processing. NLQPS is a restricted domain system, deals with the natural Language Queries on time enquiry for different modes of transportation. The system performs a shallow syntactic and semantic analysis of the input query. After the knowledge level understanding of the query, the system triggers a reasoning process to determine the type of query and the result slots that are required. The investigator tries to extract the hidden intelligent behind a Natural Language Query submitted by a user.

KEYWORDS

Natural Language Processing (NLP), Query Processing (QP), Language Model (LM), Information Retrieval (IR).

1. INTRODUCTION

A NLQPS system is an automatic system capable of processing the written natural language query like a human. The NLQPS systems can be characterized with several qualities that fundamentally arise in a Language Processing System. A Query Processing system can be domain specific, which means that the topics of the query are restricted. Often, this means simply that also the document collection, i.e., the corpus, in which the answer is searched, consists of texts discussing a specific field. This type of NLQPS is easier, for the vocabulary is more predictable, and ontologies describing the domain are easier to construct. The other type of NLQPS, open-domain query processing, deals with unrestricted topics. Hence, query may concern any subject. The corpus may consist of unstructured or structured texts. Yet another way of classifying the field of NLQPS deals with language. In monolingual NLQPS both the query and the corpus are in the same language. In cross-language NLQPS the language of the query (source language) is different from the language of the documents (target language).

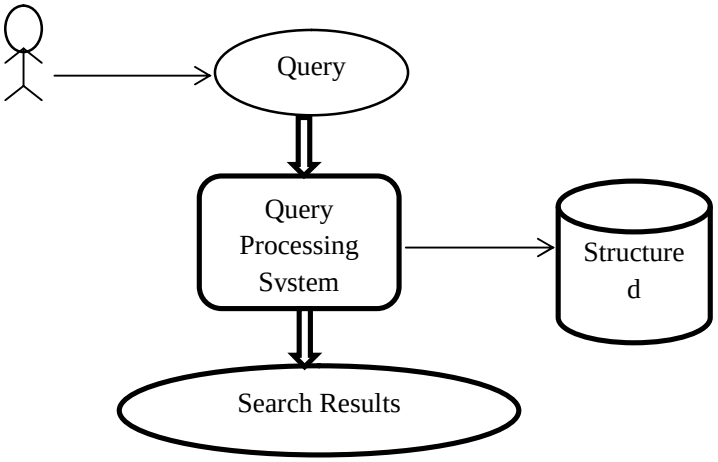
Since the early days of artificial intelligence in the 60's, researchers have been fascinated with answering natural language query. However, the difficulty of natural language processing (NLP) had limited the scope of NLQPS to domain-specific expert systems. NLQPS has been studied in NLP since 1970s with the systems like BASEBALL [13], which provides answers to query about the American Baseball League and LUNAR [11], which allowed geologists to ask query about moon rocks. In recent years, the combination of web growth, improvements in information technology, and the explosive demand for better information access has reignited the interest in QA systems.

DOI :10.5121/ijcsa.2013.3206

QA is regarded as more complex NLP application than other types of applications like information retrieval (IR) or information extraction (IE), and it is some time regarded as paramount of IR/IE. Typically QA is supported by Natural Language Processing and IE [16].

The applications that will be possible when NLP capabilities are fully realized are impressive computers would be able to process natural language, translating languages accurately and in real time, or extracting and summarizing information from a variety of data sources, depending on the users requests. Now there is clear need for improved techniques to organize large quantities of information. Applied and theoretical research in the areas of information authorship, processing, storage and retrieval is of interest to all sectors of the community. In this work the investigator's focus is strictly on the retrieval of information in response to Natural Language queries [15]. At present almost all information available in electronic form, hence an intelligent Query processing system is capable of locating information submitted by a Natural Language Query. An information agent is in control for providing the knowledge understanding of information. This searching system model includes a shallow parser for analyzing the query, a transformer for transforming syntactic structure of query to its semantic representation, a generator for generating queries on relational database from semantic model, and a constructor to construct SQL query for retrieval [1] request raised by the user. The information agent accepts the Natural language Query explores the idea of user request and transform into a structured query language query or a query to the Information retrieval system [figure 1]. To explore the actual idea of a Natural Language Query by a machine require the various levels of Natural Language Processing techniques [14].

Figure 1: Query processing system



2. RELATED WORK

The very first attempts at NLP database interfaces are just as old as any other NLP research. In fact database NLP may be one of the most important successes in NLP since it began. Asking queries to databases in natural language is a very convenient and easy method of data access, especially for casual users who do not understand complicated database query languages such as SQL. The success in this area is partly because of the real-world benefits that can come from database NLP systems, and partly because NLP works very well in a single-database domain. Databases usually provide small enough domains that ambiguity problems in natural language can be resolved successfully. Here are some examples of database NLP systems:

LUNAR (Woods, 1973) involved a system that answered queries about rock samples brought back from the moon. Two databases were used, the chemical analyses and the literature references. The program used an Augmented Transition Network (ATN) parser and Woods' Procedural Semantics. The system was informally demonstrated at the Second Annual Lunar Science Conference in 1971. [6]

LIFER/LADDER was one of the first good database NLP systems. It was designed as a natural language interface to a database of information about US Navy ships. This system, as described in a paper by Hendrix (1978), used a semantic grammar to parse queries and query a distributed database. The LIFERILADDER system could only support simple one-table queries or multiple table queries with easy join conditions. [5]

ELIZA [11] has been a classical work on natural language Query Processing mimicking natural human communication and intelligence. It uses a number of key-words and phrases to guess the type of question and for generating slots for outputs from the database. It does not use linguistic knowledge. Such a system is of limited practical use due to its limited ability to handle variety of constructs [8].There has been a lot of research on developing more practical systems catering to multiple domains [5],using multiple knowledge sources [3,4], and for cross lingual information retrieval [7].

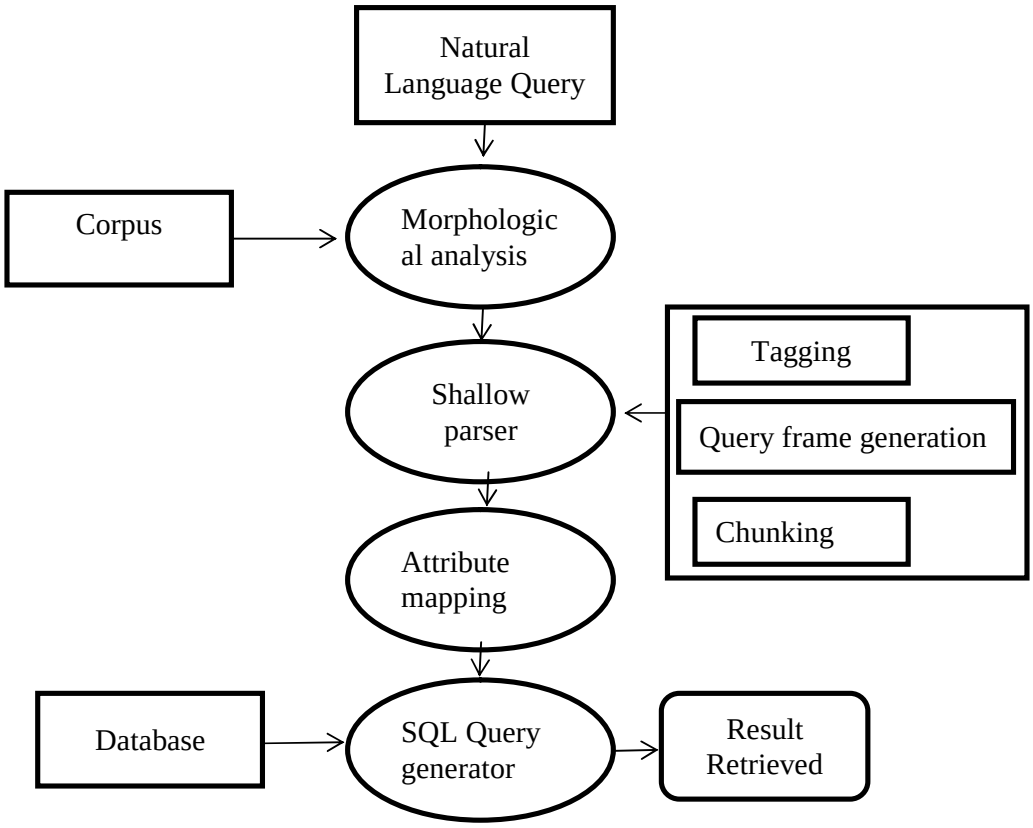
3. SYSTEM DESCRIPTION

The user specifies the query language while starting a dialogue with the system. The language specific shallow parser receives each query in the dialogue for analysis. It tags the input query morpho-semantically using the domain ontology present in the domain and linguistic models. These tagged words are assembled into chunks that represent source destination name chunks, vehicle name chunks, keyword chunks etc. The keyword chunks and sometimes information chunks are used to identify the topic of the query, i.e., the query frame. The domain model includes the words (or chunks) that occur in the input query in each language, together with the type of the chunk. It also includes the default value rules and the interpretive rules in each language. The linguistic model for each language includes the inflections, post-positions or route words and the keywords for handling the inflected words and to identify the query topic unambiguously. The user model includes the user details like name, service, designation, office and home address and the journey class frequently travelled by the user among others that are considered by the system while predicting the user intentions or making recommendations to the user. Query frame is called to generate all necessary SQL statement(s) using the information chunks. In this system, two aspects of dialogue are modelled by the DM, a generic description of how the dialogue can be interpreted by the dialogue model [12] and the representation of resulting dialogue in the dialogue history [13] that contains previous information. DM also keeps track of the anaphoric / elliptical queries from the user that constitutes the dialogue.

4. SYSTEM ARCHITECTURE

In this keyword based approach the input query statement is analyzed by the query analyzer, which uses domain ontology stored as knowledge base, generating tokens and keywords. The appropriate query frame is selected based on the keywords and the tokens in the query statement. Each query frame is associated with a SQL generation procedure. The appropriate SQL statement(s) is generated using the tokens retrieved from the input query.

Figure 2: Natural Language Processing levels on a Query



4.1. General Format, Page Layout and Margins

1. Query Interface: Designing the front end, where the user will enter the query in natural language.
2. Morphological Analyzer: The given input is broken in to token into several token.

3. Shallow Parsing: shallow parser provider the structure the structural basis for NLP queries.

    - Tagging: The individual words are tagged depending up on the specific database domain and the keywords.
    - Chunking: Chunking is a process of finding related group of word.
    - Query frame Decision: Restricting the query domain and information domain.

4. SQL Generator: The output from the parser will generate an appropriate SQL query for the given statement.
5. Data Collection: The query will interact with the database.
6. Answer Generator: The output of the given query is generated.

5. DESIGN OF INFORMATION SYSTEM AND LINGUISTIC MODEL

The database is structured and contains the information to provide the transport information service. Transport information system, database contains information about the arrival/departure time of vehicles. The aim of database management is to describe the information, in order to offer the service. The main tables used here are schedule table for each vehicle. Domain and linguistic model hold the knowledge of the world that is being talked about. Information from the domain and linguistic model is primarily used to guide the semantic interpretation of user's requests; to find the relevant items and relations that are discussed, to supply default values, etc. The knowledge presented in this domain and linguistic model is coupled to the background database system.

6. DESIGN OF THE CORPUS

The system maintains a corpus of the domain to facilitate system. For a system operating on a restricted domain this is quite obvious since it will greatly improve the disambiguation and parsing. The words that occur in the database query for enquiry system includes words describing vehicle details, and date and/or period of journey or keywords that specify the topic of the query. Hence we stored a domain dependent ontology in the knowledge base.

7. NATURAL LANGUAGE PROCESSING OF THE QUERY

In this work there is a corpus for the possible words in the transportation time enquiry system. The user query will be processed in various Natural language processing levels. For this it required the linguistic information about the reflected form of the words. Here we have conducted a detailed study on the

Malayalam inflections :- (BUO (*ill*[from])), (\Dda (*lake*[to])), (%[3(*oude*['s])) etc..

Some input querys are:-

1) #DgWKBUO<Uka *jWCU\Dda \=T*Wk വാഹനങ്ങൾ ([8T[d ?
2) #DgWKBUO<Uka *jWCU\Dda8UvJTKa/b \=T*Wk വണ്ടികൾ ([8DcT^ ?

#DgWKBUO(from alappuzha)
<Uka(From)
*jWCU\Dda(to kannur)
8UvJTKa/b (Monday)
\=T*Wk(going)
വണ്ടികൾ (vehicles)
([8DcT^ (Which)

We have considered possible postpositions
Like <Uka ([from]). (For eg: (#DgWKBUO).
[from alappuzha])), which can be used to identify the source station in the input query .We kept a list of keywords in a table in order to identify the proper query frame.

The input query statement is carried out to identifythe root words / terms. Analyzing the whole input query, the system identifies several tokens such as Train name, Station name, date and period of the day etc. and a set of keywords.

For example:#DgWKBUO<Uka *jWCU\Dda \=T*Wk വണ്ടികൾ ([8T[d ?

Here query is parsed based on spaces. Afterparsing each word, it is searched in the knowledge base until the word is found. After searching each word/term in the knowledge base, their types and semantic information are put in a list of tokens. Each token has three properties: the token value, its type and semantic information that it contains. These tokens and keywords are used to decide the proper query frame. For the above example, the tokens identified are

#DgWKBUO(from allapuza), <Uka(From)
*jWCU\Dda(to kannur),8UvJTKa/b (Monday)
\=T*Wk(going), വണ്ടികൾ (vehicles), ([8DcT^ (Which)are under keywords list.

8. QUERY FRAME DECISION

The keywords in the input query are detected. In this step, based on the tokens and keywords, we identify the appropriate query frame. Restricting the query domain and information resource, the scope of the user request can be focused. That is, there are a finite number of expected question topics. Each expected question topic is defined under a single query frame. Some query frame examples for Railway information system are fare of a journey [Fare], arrival [Arr_Time] or departure time [Dep_Time] of a train, trains between important stations [Imp_Stations], scheduled time [Sched_Time], weekly frequency of a train [Arr_Frequency / Dep_Frequency. The selection process of query frame has a great influence on the precision of the system, while there is not much likelihood of errors in other processes, such as getting the information from the dialogue history or generating SQL statement(s) from the selected query frame and/or retrieving the answer from the database and generating natural language answer from the retrieved result.

9. SQL GENERATION

Dependent upon the keyword and tokens the query frame is selected for a question, the corresponding procedure for the SQL query generation is called. For each query frame there is a procedure for SQL statement(s) generation [16]. In order to generate the SQL query, it needs the tokens generated by the query analyzer.

SQL generation procedure considers that the user provide the necessary information.

For example:-

#DgWKBUO<Uka *jWCU\Dda \=T*Wk [`3BU<W*P ([8T[d ?

Which are the train running from alappuzha to kannur?
The [Train_name] Query frame is selected.
SELECT trainName FROM train_details
WHERESource_station_Name= #DgWK destination_station_Name = *jWM;

10. ANSWER GENERATION

Once the SQL statement for an input query statement is generated, it is triggered on the database and the retrieved information is used to represent the answer.

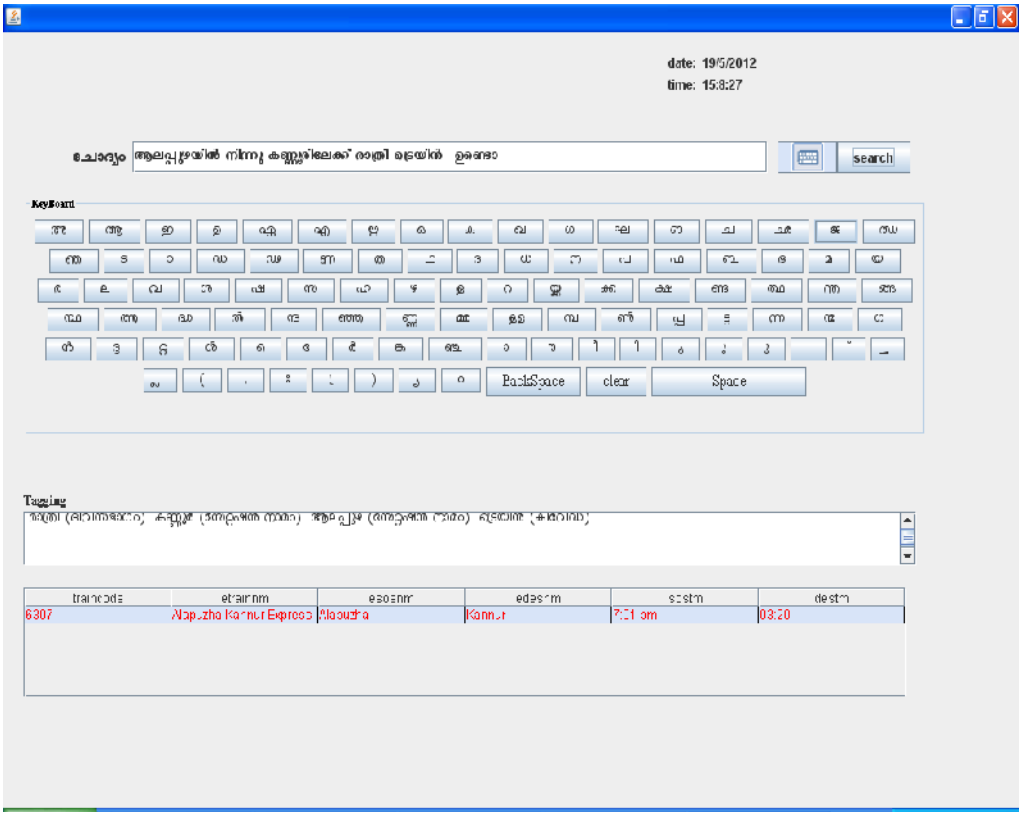


Figure 3

11. EVALUATION

To evaluate our system, we had taken queries from common people who travel by train in Kerala. They have also been told about the constraints on the nature of queries in the system. We had shown an example query accepted by the system. We evaluated our system, with and without the dialogue management.

Evaluation of system without dialogue management: The system without dialogue management means, one question, and one answer, there is no such an interaction between user and system. Here we are considering two measures for evaluating the system without dialogue management. Those are Precision and Recall.

Precision = (Number of correct outputs generated by the system / Number of outputs generated by the system) * 100.

Recall = (Number of correct outputs generated by the system / Number of natural language queries given to the system) * 100.

The system is evaluated by giving a set of 70 queries. Out of 70 queries, system has generated output for the 64 queries. Out of 64 outputs, 56 are identified as correct outputs, for the remaining 8 queries, system unable to generate the exact output.

Number of outputs generated = 64.
Number of correct output generated = 56.

Precision = (56/64)*100 = 87.50%.
Number of natural language queries given to the system = 70.
Recall = (56/70)*100 = 80.00%.

12. CONCLUSION

In this NLP system following the keyword based approach, each word need not be found in the knowledge base. Only the words that contain semantic information needs to be found in the knowledge base. By restricting the coverage of queries, our system could achieve relatively high dialogue success rate. However, for a real practical system this success rate must be improved.

The multilingual restricted domain Query Processing system separates dialogue control from the application logic, provides a portable dialogue manger and a user- friendly interface. The Dialogue manager is language independent separate modules have been developed to generate the necessary SQL statement(s) to retrieve the result, which is used in generating the natural language answer by the answer generator. Some preliminary evaluation of the system has been carried out. We are developing more robust shallow parser and the modules for the remaining query frames with dialogue management. Similar modules have to be developed for other Indian languages. The scope of the system should be extended to handle information retrieval from documents.

The system needs to be upgraded so that a user can query for railway information over phone. The speech input can be converted to textual query. This textual query can be input of our system and the textual output can be converted to speech again to answer the user. It is not realistic to assume that text-based dialogue systems can be converted into speech-based dialogue systems trivially, e.g. by adding speech recognition and synthesizer components. The focus of these systems is different, and some of the research queries, especially those dealing with the nuances of written text, are not particularly relevant in speech systems. Nevertheless, speech systems can utilize knowledge from text-based dialogue systems

REFERENCES

[1] R.K. Agnihotri, Mixed Codes and their Acceptability. In R.K. Agnihotri and A.L. Khanna (eds) Social Psychological Perspectives on Second Language Learning, New Delhi Sage Publication, 1998, 215-230.

[2] Dang Tuan Nguyen, Tuyen Thi-Thanh Do, “E-Library Searching by Natural Language Question-Answering System”, Proceedings of the Fifth International Conference on Information Technology in Education and Training (IT@EDU2008), pages: 71-76, Ho Chi Minh and Vung Tau, Vietnam, December 15-16, 2008

[3] Tej Bhatia and William Ritchie, Bilingual Language Mixing, Universal Grammar, and Second Language Acquisition. In William Ritchie and Tej Bhatia (eds) Handbook of Second Language Acquisition, San Diego: Academic Press, 1996, 627-688.

[4] Claire Cardie, Vincent Ng, David Pierce, and Chris Buckley, Examining the role of statistical and linguistic knowledge sources in a general-knowledge question answering system, In Proceedings of the Sixth Applied Natural Language Processing Conference (ANLP-2000), 2000, 180-187.

[5] Jennifer Chu-Carroll, Krzysztof Czuba, John Prager, and Abraham Ittycheriah, In Question Answering, Two Heads are Better Than One, HLT/NAACL-2003.

[6] Hoojung Chung, Young-In Song, Kyoung-Soo Han, Do-Sang Yoon, Joo-Young Lee, Hae-Chang Rim and Soo- Hong Kim, A Practical QA System in Restricted Domains, Workshop on Question Answering in Restricted Domains at ACL 2004, Barcelona, Spain.

[7] Mitrovic, A.A knowledge-based teaching system for SQL, University of Canterbury, 1998. Moore, J.D. "Discourse generation for instructional applications: making computer tutors more like humans", in Proceedings AI-ED, pp.36-42, 1995.

[8] ELF Software CO. Natural-Language Database Interfaces from ELF Software Co, cited November 1999, available from Internet: http://hometown.aol.comJelfsoft!

[9] R.M.K. Sinha, An Engineering Perspective of Machine Translation: AnglaBharti-II and AnuBharti-II Architectures, Invited Paper, Proceedings of International Symposium on Machine Translation, NLP and Translation Support System (iSTRANS- 2004), November 17-19, 2004, Tata McGraw Hill, New Delhi.

[10] R.M.K. Sinha and G.C. Pathak, A heuristic based question answering system in natural Hindi, IEEE-SMC International conference, Delhi-Bombay, Dec.30, 1983- Jan.7,1984, 1009-13.

[11] Akama, S. (Ed.) Logic, language and computation, Kulwer Academic publishers, pp. 7-11, 1997

[12] Joseph Weizenbaum, ELIZA—a computer program for the study of natural language communication between man and machine, Communications of the ACM, 9(1), 1966, 36-45.

[13] Huangi, GuiangZangi, Phillip C-Y Sheu "A Natural Language database Interface based on probabilistic context free grammar", IEEE International workshop on Semantic Computing and Systems 2008

[14] Akama, S. (Ed.) Logic, language and computation, KulwerAcademic publishers, pp. 7-11, 1997.

[15] Hendrix, G.G., Sacerdoti, E.D, Sagalowicz, D., Slocum, J. "Developing a natural language interface to complex data", in ACM Transactions on database systems, 3(2), pp. 105-147, 1978.

[16] Joseph, S.W, Aleliunas, R. "A knowledge-based subsystem for a natural language interface to a database that predicts and explains query failures", in IEEE CH, pp. 80-87, 1991.

[17] Mitrovic, A.A knowledge-based teaching system for SQL, University of Canterbury, 1998. Moore, J.D. "Discourse generation for instructional applications: making computer tutors more like humans", in Proceedings AI-ED, pp.36-42, 1995.

[18] Suh, K.S., Perkins, W.C., "The effects of a system echo in a restricted natural language database interface for novice users", in IEEE System sciences, 4, pp. 594- 599, 1994.

[19] Whenhua, W., Dilts, D.M. "Integrating diverse CIM data bases: the role of natural language interface", in IEEE Transactions on systems, man, and cybernetics, 22(6), pp. 1331-1347, 1992.

[20] Dan Klein, Christopher D. Manning: Corpus-Based Induction of Syntactic Structure: Models of Dependency and Constituency. ACL 2004: 478-485.