International Journal of Interdisciplinary Innovative Research & Development (IJIIRD)
ISSN: 2456-236X
Vol. 03 Issue 01 | 2018

Bilingual Query Processing System using Software Agents

Dr. Nilesh Shelke
Assistant Professor, Computer Science and Engineering Department, Nagpur, MH,India

ABSTRACT

So far, there has been an impressive amount of research on natural language interfaces (NLIs), i.e. on interfaces allowing users to interact with a certain information system in natural language. Day by day the communication gap between the computer and a human is getting minimized. Natural Language Interfaces to Databases (NLIDBs) is one of the mechanisms to pull off this goal. In NLIDBs the question is asked in simple daily life human language and the answer is given in the same language. This research paper is related to work that efficiently maps a natural language query, entered in English, to an SQL (Structured Query Language) statement and converting the result into any other language like Hindi. Much of the work has been done in this field, but our focus is to use software agents for it.

***Keyword:** - NLP, Agent, Transliteration, Tokenization*.

1. INTRODUCTION

"Why aren't computers easier to use?" inquired the unsuspecting beginner computer user. Such an easily posed question has many long and complicated answers. 'Why don't you understand!" demanded the frustrated computer abuser. This is situation that computer programmers and researchers spend their lives studying and trying to prevent.

Use of software agents in the field of Natural language processing is very less [1].They behave like human intelligently, autonomously, cooperatively, and socially to solve problems or to support human users [2]. The goal of this project is to study the utility of the agents in the real time environment.

The system we have developed is a multi-agent open system where agents are capable of flexible autonomous action in order to meet their design objectives. Each agent exhibits pro-activity, sociability and learning capability. Pro-activeness is attributed to the goal directed behavior of each agent. Sociability is referred to the interaction with other agents to get its objective done and learning ability attributed to the capability to adapt to user's language preferences and new languages. The database considered in this study is Movie database created from the information of bollywood films on internet. The user can give his query to the system as if he delegates a human being for information gathering. There is no rigid syntax for giving the query. The system uses NLP techniques to extract meaning of the query and retrieves information from data stores and presents the information to the user in their own native language.

2. SYSTEM ARCHITECTURE

In a broad sense, an *agent* is any program that acts on behalf of a (human) user. Agent is "someone (or something) that has the ability, power or authority to act", When applied to software "a program that does something on behalf of the user" [1].

2.1 User Interface Agent

It accepts typed natural language queries from the user and presents response to the queries in Hindi and English. Query can be input either through the keyboard or through the onscreen keypad with the help of touch pen.

2.2 Server Agent

The sever agent is a blackboard server agent that is responsible for coordinating agent communication and control and for providing a global data store to its client agents. It maintains a registry of agent service and data declarations. All communication between client agents must pass through the server Agent.

2.3 Parser Agent

Parsing Agent gives us the ability to extract certain keywords or indicators in the order they occur in a sentence. Using these keywords and indicators and their order in the sentence one can do the actual SQL query generation. So unlike the traditional methods which involve detailed syntactic and semantic analysis our method is very computationally inexpensive. It does require the building of Preprocessing before hand, but once that is done the process of feeding a string and extracting keywords is very inexpensive. Once we have the keywords, indicators and their relative ordering we input that information to the SQL Agent.

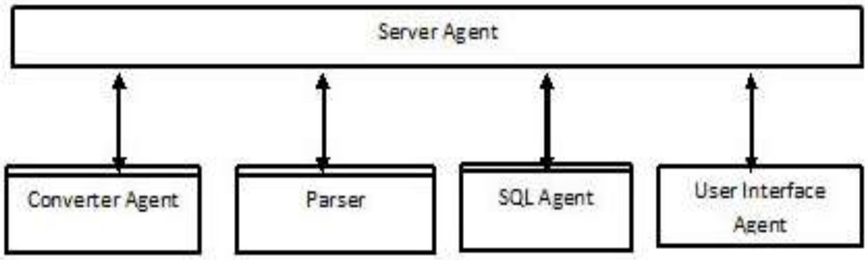


Fig -1Architecture of the System

Following are the steps of Parsing Agent:

1-Tokenization: As a first step in processing a query, it has to be determined what the processing tokens are. One of the simplest approaches to tokenization defines word symbols and inter-word symbols. All characters that are no letters and no digits are considered to be inter-word symbols. The inter-word symbols are ignored during this phase, and the remaining sequences of word symbols are the processing tokens. As a result it is not possible to search for punctuation marks like for instance hyphens and question marks.

2-Stop word removal: Stop words are words with little meaning that are removed from the query. Removing stop words for conceptual reasons can be done by using a stop list that enumerates all words with little meaning.

3-Morphological Normalization: Morphological normalization of words in queries is used to find words that contain morphological variants of the original query. Morphological normalization can be achieved either by using a stemmer or by using dictionary lookup. For Example, what are the names of ………? Here dictionary lookup should give the idea that the root word of names is name.

5-Synonym Normalization: Synonyms words might also be conflated to one processing token during indexing and automatic query formulation. For Example the query 'What are the movies of Amitabh Bachchan' has the same meaning as 'What are the Films of Amitabh Bachchan?' or What are the stars, casts or actors are equivalent words.

So the first step in our method is extracting the keywords from the English sentence. Given information about the database we generate that will do the task of extraction. The keywords that we extract are words that refer to table names, field names, operators etc.

Figure 2 contains some sample sentences inputted to the parsing agent and their corresponding sample output.

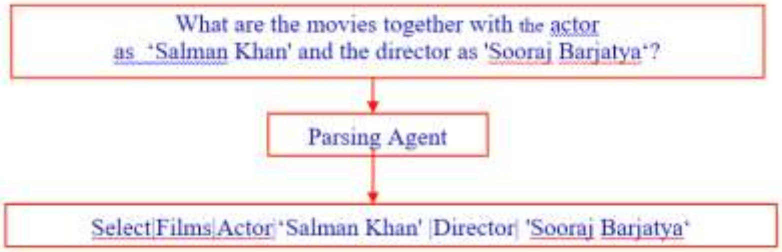

What are the movies together with the actor as 'Salman Khan' and the director as 'Sooraj Barjatya'?
Parsing Agent
Select|Films|Actor|'Salman Khan' |Director| 'Sooraj Barjatya'

Fig -2 Sample sentence to the Parsing Agent and its output

2.4 SQL Agent

Once we have all the keywords and their relative order from the preprocessing, this information is given to the SQL Agents. It should be noted that SQL Agent contains information about the database such as the table names and the field names. The SQL Agent looks at the final SQL query as broken down into two components. The Objects component and the optional Conditions component. The SQL Agent tries to determine if there is a Conditions component by looking for its corresponding indicator, which is whose. If it does exist, it splits the input string into two components, consisting of words that occur before the indicator and after the indicator. The logic behind this is that most human queries are structured in a way similar to the ones mentioned in the examples in Figure 2. However there are sentences that don't adhere to this rule. In such cases we can set up the system so that it iterates back and forth between the users to see if there is a similar sentence that does match the required structure based on some pattern matching.

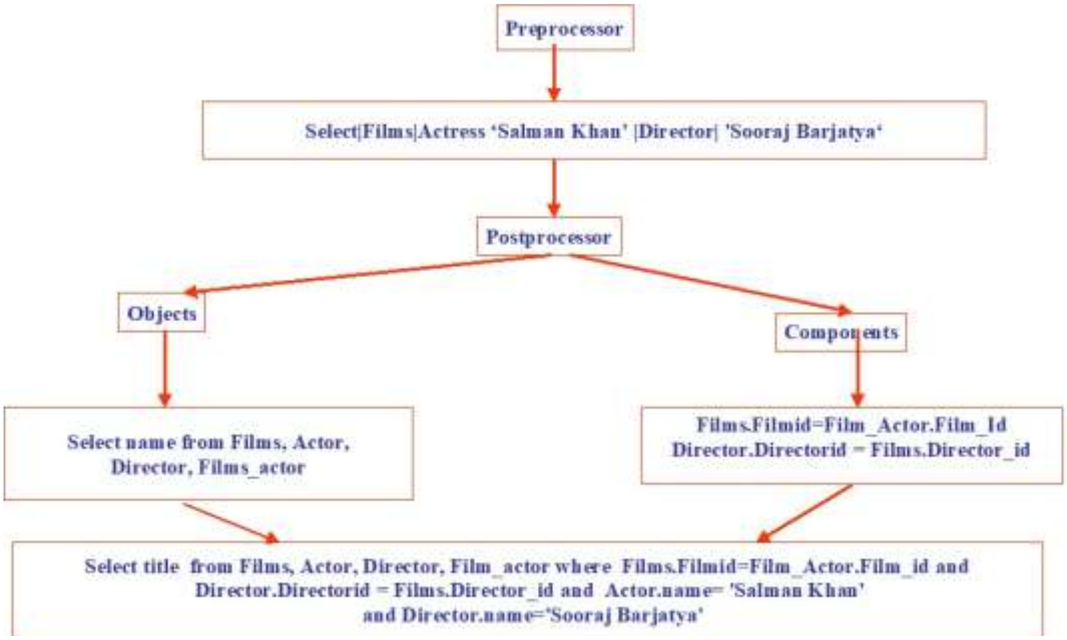

Preprocessor
Select|Films|Actress 'Salman Khan' |Director| 'Sooraj Barjatya'
Postprocessor
Objects
Components
Select name from Films, Actor, Director, Films_actor
Films.Filmid=Film_Actor.Film_Id
Director.Directorid = Films.Director_id
Select title from Films, Actor, Director, Film_actor where Films.Filmid=Film_Actor.Film_id and Director.Directorid = Films.Director_id and Actor.name= 'Salman Khan' and Director.name='Sooraj Barjatya'

Fig. 3 Working of SQL Agent

2.4.1 Objects component

Irrespective of the presence of Conditions component the Objects component will always exist. So once we have that, the SQL Agent, iterates through it to find all the fields that are mentioned in the component. Once it has this it checks to see what tables a field is associated with based on the information. In case if it finds a field that is associated with multiple tables such as name, or musician, then it iterates through the table mentioned in the whole string and does elimination based on the presence of tables in the string. In case both (or more) tables are in the string then it associates the table that is relatively closer to the field in the string. Once it has the fields and the tables associated with them it makes a Semi Query in the format:
SELECT [tablenames.fieldnames] FROM [tables]

2.5 Converter Agent:

The problem can be stated formally as a sequence labeling problem from one language alphabet to other. Consider a source language word $x_1x_2....x_i..x_N$ where each xi is treated as a word in the observation sequence. Let the equivalent target language orthography of the same word be y_1, $y_2...y_i..y_N$ where each y_i is treated as a label in the label sequence. The task here is to generate a valid target language word (label sequence) for the source language word (observation sequence).

x_1 ——————— y_1
x_2 ——————— y_2
. ————————- .
. ————————- .
. ————————- .

x_N ———————— y_N

Here the valid target language alphabet (y_i) for a source language alphabet (x_i).

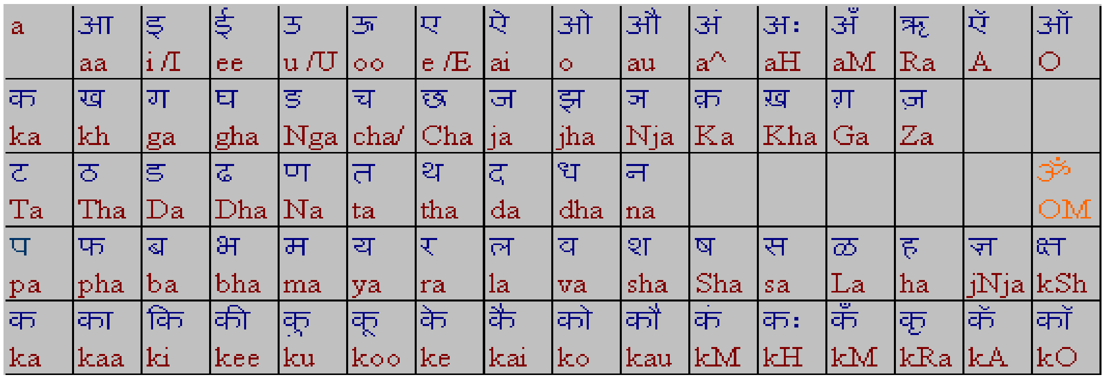
Our Experimental Work:

Following different options were considered for transliterating the words of Hindi into English.

- Shrilipi (which is a software mostly used for printmedia related work). It is licensed, Platform dependent and application dependent.
- Government related software's ism one can write keeping the same font as ASCII values are given same to English and rest is for Hindi. License Platform and application dependent. Encrypt for the Hindi letters.
- Script based font supported by windows or other application developers. Platform and application dependent
- Krutidev same keyboard standard layout freeware.

Out of these using freeware Krutidev was the best option

Fig. 4 shows how English letters can be made equivalent to the Hindi words.



a	आ aa	इ i /I	ई ee	उ u /U	ऊ oo	ए e /E	ऐ ai	ओ o	औ au	अं a^	अः aH	अँ aM	ऋ Ra	ऍ A	ऑ O
क ka	ख kh	ग ga	घ gha	ङ Nga	च cha/	छ Cha	ज ja	झ jha	ञ Nja	क़ Ka	ख़ Kha	ग़ Ga	ज़ Za		
ट Ta	ठ Tha	ड Da	ढ Dha	ण Na	त ta	थ tha	द da	ध dha	न na						ॐ OM
प pa	फ pha	ब ba	भ bha	म ma	य ya	र ra	ल la	व va	श sha	ष Sha	स sa	ळ La	ह ha	ज्ञ jNja	क्ष kSh
क ka	का kaa	कि ki	की kee	कु ku	कू koo	के ke	कै kai	को ko	कौ kau	कं kM	कः kH	कँ kM	कृ kRa	कॅ kA	कॉ kO

Fig. 4 Matrix for Transliteration of English letters to Hindi

Algorithm used for Transliteration:

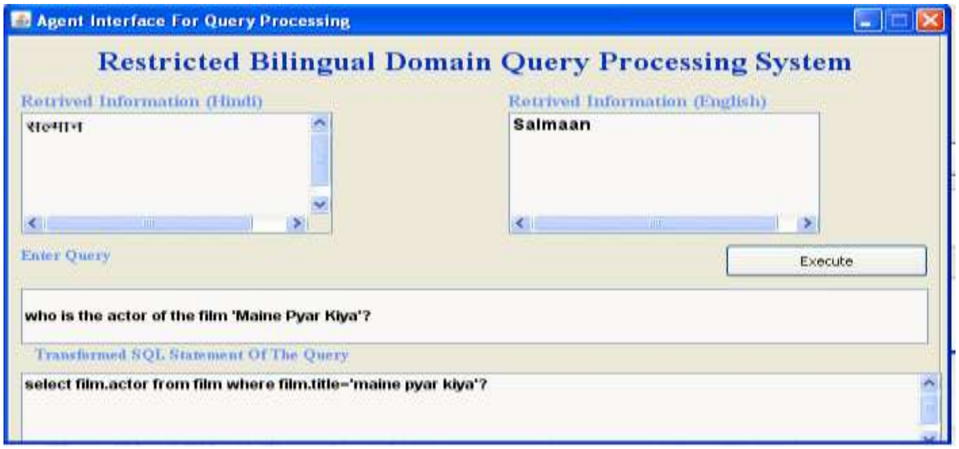
```
For all the array of words in Quotation
  For each word
        If the character is not vowel with then
                Read the further character till you  get the consonant.
                Make a combination of this leaving the just occurred consonant and search it in the given matrix
                as in fig 1.
        Else
        If Vowel then
                Read the further character till you  get the consonant.
                Make a combination of this leaving the just occurred consonant and search it in the given matrix
                as in fig 1.
```

Example for the Hindi Movie "Kaun":

Select musician from movie where title='kaun'

Select musician from movie where title='dkSu'

Filmid	Title	Musician	Year	Directorid	Producerid
1	eSus I;kj fd;k	vkj fM cjeu	1992	1	1
2	dkSu	vkj fM cjeu	2000	1	1
3	fnyokys nqYguh;k ys tk;saxs	, vkj jgseku	2005	2	2



Restricted Bilingual Domain Query Processing System

Retrived Information (Hindi)

सल्मान

Retrived Information (English)

Salmaan

Enter Query

who is the actor of the film 'Maine Pyar Kiya'?

Transformed SQL Statement Of The Query

select film.actor from film where film.title='maine pyar kiya'?

Fig. 5 Screen shot from the research work

3. CONCLUSION & FUTURE WORK

The advantage of this system is that the user can query the data store in English without knowing the complexity of the database structure and location of the database. The agents involved in the system are adaptive to the user language and the usage style. It is the server agent who is planning and coordinating the sequence of tasks involved in query processing. It adds more value to the result and can be effectively used by government bodies for resource. The MOVIES database NLP system is currently capable of handling simple queries, standard join conditions, aggregate functions and GROUP BY clauses. Because not all forms of SQL queries are supported, further development would be required before the system can be called robust natural language interface.

4. REFERENCES:

[1] Drazen Draskovic, Vidor Gencel, Slavko Zitnik, Marko Bajec, Bosko Nikolic, “A software agent for social networks using natural language processing techniques”, published in the procedings of 24th Telecommunications Forum (TELFOR), IEEE publication, pp. 45-62, 2017.

[2] Latha Subramainan, Mohd Zaliman, Mohd Yusoff, Moamin A Mahmoud, “A classification of emotions study in software agent and robotics applications research”, proceedings of International Symposium on Agents, Multi-Agent Systems and Robotics (ISAMSR), IEEE publication, IEEE publication, 2016.

[3] Agung Fatwanto, “Translating software requirements from natural language to formal specification”, procedings of IEEE International Conference on Computational Intelligence and Cybernetics (Cybernetics.Com), IEEE publication, 2013.

[4] Sebastian Weigelt, Walter F. Tichy, “Poster: ProNat: An Agent-Based System Design for Programming in Spoken Natural Language”, IEEE/ACM 37th IEEE International Conference on Software Engineering, IEEE publication, 2016.

[5] J. Guadalupe Ramos Díaz; Nicolás Jasso García; María Estela Romero, “Mobile agent for the building of automatic textual reports from web queries”, IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC), IEEE publication, 2016.