International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor:6.887
Volume 5 Issue IX, September 2017- Available at www.ijraset.com

Three Phase Synthesizing Of Nlp and Text Classification for Query Generation

Amisha Shingala[1], Anjali Jivani[2]
[1]*Dept. of MCA, SVIT, Gujarat Tecnological University*
[2]*Dept. of CSE, The M. S. University of Baroda*

Abstract*: With the great upsurge of usage of Whatsapp, Facebook, Twitter, etc. as well as firing queries to Google for every answer that a person requires, the trend of using Natural languages other than English using English alphabets has become very common. For example instead of writing 'Kem Chho' in Gujarati script we find it easy to write using English alphabets. Our attempt in this work has been in trying to convert such Gujarati and Hindi scripts which are actually written using English alphabets to be converted to Structured Query Language (SQL). This has been done by designing initially corpus of these languages, and with every new natural language query that is fired, trying to classify it using the pre-defined classifiers and using a heuristic approach increasing the scope and size of the corpus. In this a person who is comfortable in his native language can get information from our database which right now has been developed for Gujarati and Hindi language.
***Keywords*: NLP, text mining, text categorization, information retrieval, query generation**

I. INTRODUCTION

The main idea behind designing this three step approach was to make accessing of information to individuals with limited knowledge of the English language easy as all they would have to do is write a vernacular query using English alphabets. This approach has been has been termed as Synthesizing the Natural Language Processing (NLP) and Text Classification on query generation. We have tried to generate classifiers based on the kind of the query that is asked by a user which is converted to SQL by our system. The classifiers have been predefined as per the nature of the query and every new question asked by a user gets classified according to the class it is closest to. It is a heuristic approach which allows the classifiers to grow as per every new query asked and equivalent SQL is generated.

II. THE THREE STEP PROCEDURE

The three step procedure for the query generation of queries fired in Natural Language but using English alphabets is described in detail below.

A. Language Identification

It is important and the most initial part of the process to identify first the language in which the query is to be interpreted. Over here only two languages, Hindi and Gujarati are being considered. An initial corpus of both these languages has been created keeping the most commonly used words in the lexicon. Fig. 1 displays the algorithm for identifying the language.

```
Identify language from user input sentences
Function Search_language(user query)
Input: user query UQ
Returns: query LQ
[Connect to MySQL database and retrieve language keyword table LKT]
Get all tokens from UQ by using StringTokenizer method
For (all tokens of user query UQ) do
        Compare UQ with LKT words
        If match found then
                Store user query LQ in language category table
                Store date and time and increment the counter
        End If
End For
        Return LQ
```

Fig. 1 Identify the language

B. Query Existence Determination and Generation

Once we identify the language, LQ, then we have to check for the existence of user query. A user can access and manipulate query which is stored in the language knowledge base at any time as a template from Graphical User Interface (GUI). User can also select same template and without processing the linguistic analysis, can view the query result.

Moreover the user can also edit the existing template and compose a new query. Keeping user search history and providing a query template preserves the user's prior search effort, and gives a quick starting point when he/she needs to create new queries. The below given Fig. 2 depicts the method for checking the existence of the user query.is important and the most initial part of the process to identify first the language in which the query is to be interpreted.

```
Function Search knowledge (language query LQ)
Return: flag
Store  User language Query LQ into HQ

[ Connect to MySQL database and retrieve Knowledge base HT]

Do while (HT not null)

        Compare HT query with HQ
        If match found then
                Set flag =1   // search found in knowledge base
        Else
                Set flag =0  // search not found in knowledge base
        End If
End do
Return flag
```

Fig. 2 Existence of Query

Once the finding of the existence of the query is complete, as per the result the next step is the SQL query generation. There are two possibilities in the above step – either the flag is 0 or its 1. If its 1 it means a similar linguistic query exists in the corpus for which the SQL query is already generated. We simply need to execute the SQL and display the result to the use.

However if the flag is 0, then we have to process the query and generate the SQL. This is the challenging part in which we have to perform the Linguistic analysis containing the related syntax and semantic analysis.

The natural language input first undergoes a pre-processing phase in which it identifies the domain that pertains to the input query. For this, it tokenizes the input, performs morphological analyses of the words and looks them in lexicon dictionary to retrieve their syntactic and semantic properties which was discussed in our paper [1].

The pre-processing of the input query includes: (a) word based n-gram generation and its conversion into base words (c) spelling check (d) identifying domain, and (e) knowledge reuse. The context can be resolved by identifying the domain class in form of <database name> <domain name>< key terms><attribute terms>. The next step is the lexical analysis and then the syntactic analysis.ZThe Stanford parser syntactically parser the question by constructing set of rules and generate POS tagging which can be used to identify proper noun, adjective, verb, etc and also generate a parse tree as discussed in our paper[2], which can be used by semantic interpreter to transform into intermediate query using semantic rules.

The intermediate logic query generated by semantic interpreter, does not specify how to search the database to retrieve specific information. In order to retrieve, desired output required by user, the intermediate representation of query can be converted to some database query language which can be in form of any database system. The mapping of database specifies how intermediate query forms a logic predicate. The approach used here is to link logic predicate to SQL statement.

C. Automatic Classification

If the step mentioned above, the incoming query does not exist in the corpus of any of the languages then it is important to first identify to which class – in this case which Natural Language it is closest to.

To classify this new incoming query, the k Nearest Neighbor algorithm has been used. It is very crucial to first go through all the pre-processing steps like tokenization and stop words removal first. For each Hindi and Gujarati, a lexicon of stop words has been prepared.

The next phase deals with finding the Cosine Distance using which the new incoming query can be classified to its appropriate domain.

III.RESULT ANALYSIS

This three step procedure deals with two Natural Languages making query generation a lot more simpler and exciting for vernacular users as already mentioned before. Fig. 3 and Fig. 4 are the screen shots of the proposed algorithm which is implemented in JAVA using JDBC for Student repository.

Below given screen shots show sample questions in Hindi and Gujarati and the SQL queries generated against them. Fig. 3 is for Hindi language and Fig. 4 is for Gujarati language.

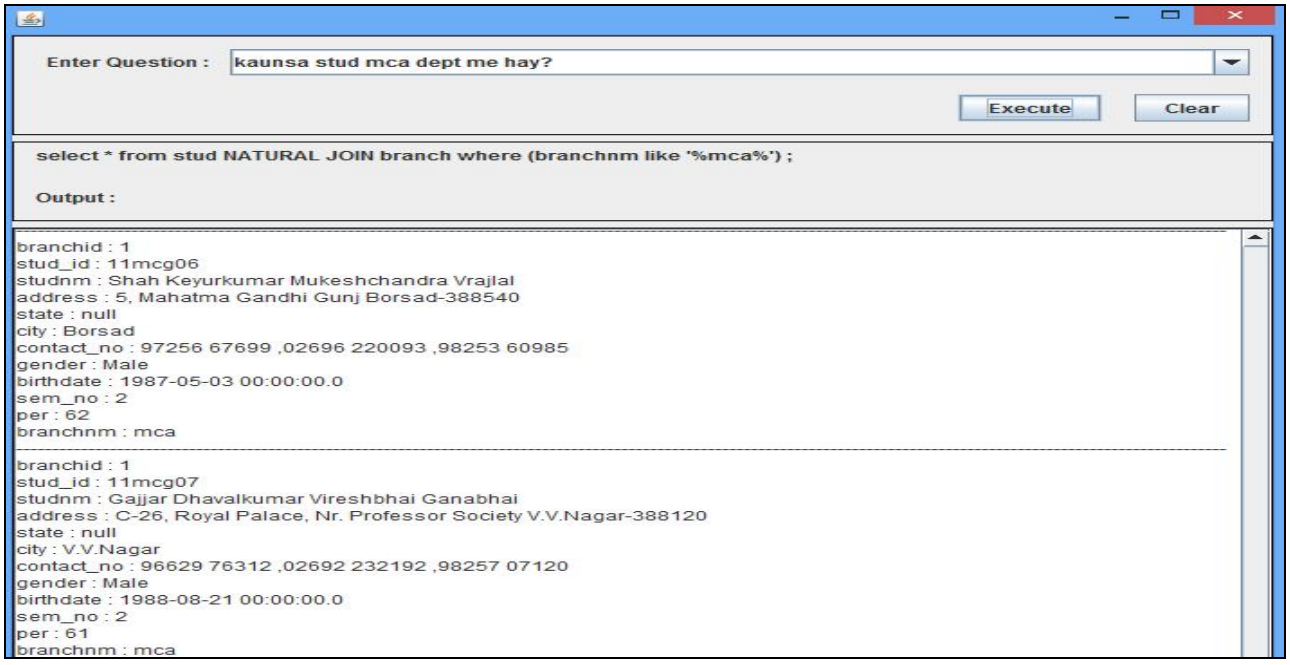


Fig. 3 Query for Hindi language

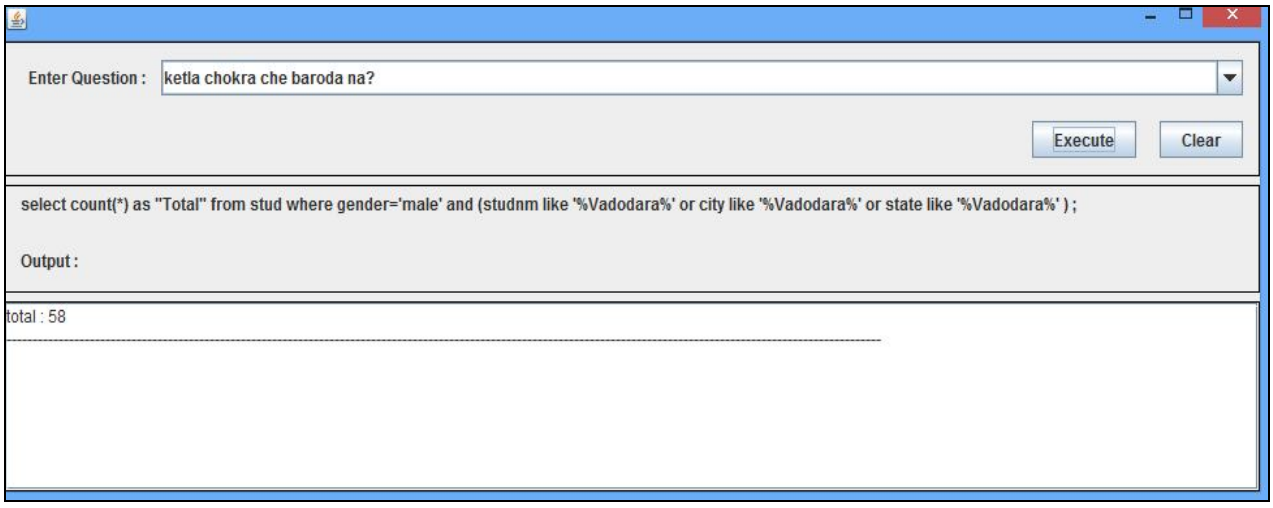
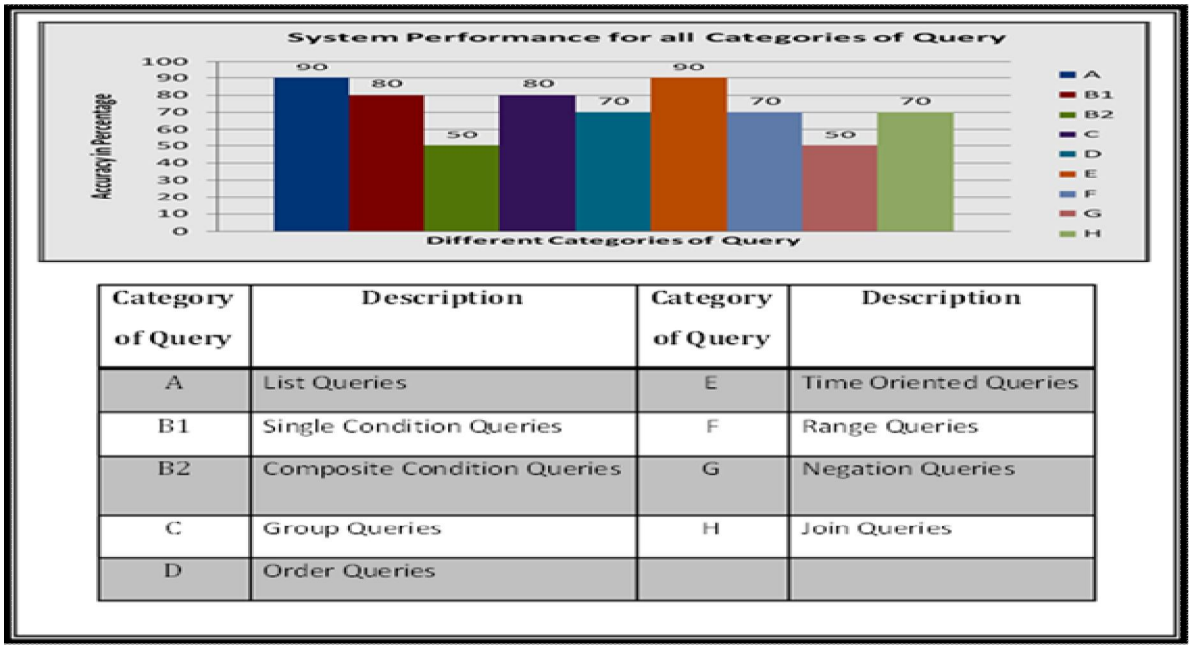


Fig. 4 Query for Gujarati language

Similarly all the queries, which are fired, the effectiveness of the category wise question in a system is measured in terms of accuracy measurement as shown in Fig. 5.



Category of Query	Description	Category of Query	Description
A	List Queries	E	Time Oriented Queries
B1	Single Condition Queries	F	Range Queries
B2	Composite Condition Queries	G	Negation Queries
C	Group Queries	H	Join Queries
D	Order Queries		

Fig. 5 System Performance for all categories of Queries

It has been observed that when the queries are simple and predictive, the output is accurate. The way Stanford Parser generates the parse tree and gives the detailed Part of Speech (POS) details for English language, there is the Shallow Hindi Parser which has been developed by the Consortium of Institutions like IIIT Hyderabad, IIT Mumbai, CDA, etc. For the Gujarati language there are no such parsers which give accurate results.

In both the cases however the challenge is converting the Natural Language query to English and then using the Stanford Parser. Classifying the query in its appropriate class of Natural Language has been done over here.

IV. FUTURE WORK

As mentioned above the future work deals with designing an appropriate and correct parser for the Gujarati language as well as designing a corpus for both the languages wherein appropriate conversions would be available for the English language. This work still needs a detailed and refined effort to get a good accuracy ratio.

REFERENCES

[1] Amisha Shingala, Rinku Chavda & Paresh Virparia, “Natural Language Interface for Student Information System”, Journal of Pure and Applied Sciences, Vol. 19:41-44, ISSN: 0975 – 2595, 2011.

[2] Amisha H. Shingala, Ms. Anjali Jivani and Dr. Paresh V. Virparia, Research paper on Multi-Liaison Algorithm, published in International Journal of Advanced Computer Science and Applications (IJACSA), ISSN No: 2158-107X(Print); 2011 Impact Factor : 1.187, volume 2, issue 5, May 2011, published by www.ijacsa.thesai.org.

[3] Ann Taylor (2000), The Penn Treebank: an overview, Chapter 1, university of York, UK, http://www.cis.upenn.edu/treebank. Alessandro Moschitti (2003), Ph.D thesis on Natural Language Processing and Automated Text Categorization: A study on the reciprocal beneficial interactions, May 8, 2003, University of Rome,Tor Vergata.

[4] Androutsopoulos, 1; Ritchie & Thanisch P, “MASQUE/SQL- An efficient and portable Natural Language Query Interface for Relational Database”, Proc of sixth International Conference on Industrial & Engineering Applications of Artificial Intelligence & Expert System, Edinburgh, 1993.

[5] Axita Shah, Dr. Jyoti Pareek, Hemal Patel, Namrata Panchal “NLKBIDB - Natural Language and Keyword Based Interface to Database” 978-1-4673-6217-7/13/$31.00_c 2013 IEEE.

[6] A.R.FALLE1, e.t.al.,’ Knowledge Extraction from Database using Natural Language Processing’, International Research Journal of Engineering and Technology (IRJET), Volume: 04 Issue: 04 | Apr -2017, e-ISSN: 2395 -0056.

[7] Cecily Heiner and Joseph L. Zachary,(2009), Improving Student Question Classification, Journal of Educational Data Mining 2009.

[8] Dushyantsinh Rathod, ‘Smart Two Level K¬Means Algorithm to Generate User Pattern Clustering’ ICTIS 2017.

[9] Garima Singh, Arun Solanki, An algorithm to transform natural language into SQL queries for relational databases Published online 1 September 2016, Selforganizology, 2016, 3(3): 100-116.