Nat. Volatiles & Essent. Oils, 2021; 8(4): 11651-11661

NVEO
Natural Volatiles &
Essential Oils

The Process Of Converting Natural Language To Sqlquery

Princy S Ajit[1] , Rohini V[2] , Kirubanand V B [3] , Balamurugan E[4] , Vinay M[5]

[1,2,3,5],CHRIST (Deemed to be University), Bangalore, India

[4] University of Africa, Toru-Orua, Nigeria

Abstract:

Natural language processing (NLP) is the abilityofacomputerprogramtounderstandhumanlanguage asitistypedinorspoken.Insimplerterms,itistheprocess of a computer understanding/interpreting natural language. Some of the applications include language translator, chatbots,voiceassistantsetc.Oneapplicationthatwillbethe main focus here is converting natural language to a SQL query.

Structured Query Language(SQL) is a powerful tool for managing data held in a relational database management system. To retrieve or manage data, users have to enter the correct SQL Query for which they need to have a proper understanding of it. To overcome this we look into the process of converting the Natural Language Query to SQL query.

Keywords: structuredquerylanguage,naturallanguageprocessing, database.

Introduction

When dealing with a typical database framework, to recover information from the database we need to know about the SQL to recover definite information from the database. Some of the types of SQL queries which will help in information retrieval are the simple select query, query with a where clause, aggregation functions, joins, having and group-by clauses. However, everybody doesn't have proper knowledge about the SQL Query language. To overcome this, there are frameworks (or systems) that can create SQL Queries from a regular query/question using Natural Language Processing (NLP).

Natural language processing (NLP) is the ability of a computer program to understand human language as it is typed in or spoken. In simpler terms, it is the process of a computer understanding/interpreting natural language. This helps novice users to easily get required contents without knowing any complex details of SQL languages. Our aim here is to understand the process behind this kind of transformation/conversion by means of building such a system.

Literature Review

The general process involves the use of semantic grammar for the conversion. There are four main steps or levels involved here, and these steps are known as Levels of Language, also known as the Synchronic Model of Language.[9]

The system will accept users' queries in natural language as an input(text, voice). Then it considers each word in the user query as a token (Morphological analysis). The tokens from the query clause are compared with clauses already stored in the dictionary and mapped to the most relevant one (Lexical Analysis). The dictionary needs to be constantly updated. Then the algorithm scans the tokens and tries to find attributes and tables present in the query and checks the dictionary of tables and attributes(Syntactic Analysis). Finally, the input query is checked for words which imply certain conditions and accordingly the where clause with relational/logical operators along with aggregate functions are framed (Semantic analysis) to formulate the final query. Confusion matrix and some other metrics are used to evaluate the accuracy[9][2].

In addition to the above steps, it may also be necessary to perform certain transformations/preprocessing to the data before applying any of the steps. The different types of preprocessing that could be applied are lowercase conversion, tokenization, escape word removal, part of speech tagging, bigram and trigram creation. With the help of the part of speech assigned to a token, it will be helpful in determining the attributes, tables, and condition- based words.[1]

Ambiguity in the words of the input query, formation of complex SQL
queries which could also include subqueries(query within a query), having a discourse knowledge in which immediately preceding sentences could affect the interpretation of the current sentence are some of the other essential considerations when making such a system.[3]

In some systems, the generated SQL query is analyzed on a larger extent to measure the semantic levels of the knowledge extracted, gained and attained, with the help of common evaluation measures such as Recall(the ability of the system to present all relevant words) and Precision(the ability of the system to present only relevant words).[4]

Certain systems are only capable of handling simple SQL queries. Whereas some other systems also focus on the creation of complex queries which includes nested queries with more than two-level depth, queries with aggregate functions, having clause, group by clause and co-related queries which are formed due to constraint on aggregate function.[5]

A model known as fr2sql deals with querying a database by means of a question which is the reason for them to only consider select queries.Around 7 lists are created each for Select, From, Join, Where, Groupby, Having, Orderby clauses. As per the user query the required tokens get added to each of these lists and the final query is simply the combination of the contents of all these lists.[11]

Some traditional systems follow the sequence-sequence model(A deep neural network model) to help in the conversion. This model is good at expressing parallel data. Here the distributed representation of table is utilized and the sequenced SQL is fed as input parallelly with the natural language query data expressing the same semantics Such systems have achieved a good rate of accuracy.[10]

But there are some limitations to these systems. Such systems have considered the order of SQL query to be significant(conditional order). But there might be some queries whose structure/order may be different but will generate correct results. Also,

such models do not leverage/exploit the full structure of the SQL query.[6]
A sequence-sequence model does not consider this aspect and gives priority to the order which may give lesser accuracy results even when the solution generated is correct. To overcome this, a policy gradient (Reinforcement Learning) is applied to this model and based on the resultant sql query generated a reward is assigned on the basis of the validity of the query and the result generated after its execution on the database. Based on this score decisions are taken to assess if the query generated is correct or not.[6] The Seq2SQL model (which uses a deep neural network) follows the above approach in its process which begins with the network classifying an aggregation operation for the query(if required), then it points to a column in the input table corresponding to the SELECT column. Finally, the network generates the conditions for the query using a pointer network, and here the policy gradient as mentioned above is applied.[6]
The SQLNet model helps in overcoming the order problem without the help of reinforcement learning. The issue with reinforcement learning is that improvement from it is limited. This model employs a sketch-based approach, where the sketch contains a dependency graph so that one prediction can be done by taking into consideration only the previous predictions that it depends on. In addition, a sequence-set model as well as the column attention mechanism(to synthesize the query based on the sketch) is proposed.[7]
The most recent developments in this area is the value-light and value-net model. The main idea behind this model is to use not only metadata information about the underlying database but also information on the base data as input for the neural network architecture. In particular, a novel architecture sketch is employed to extract values from a user question and come up with possible value candidates which are not explicitly mentioned in the question. Then a neural model is used based on an encoder-decoder architecture to synthesize the SQL query.[8]

Research Hypothesis/Model/Framework

The implementation/procedure explained in the next section, at its core is adopted from the ln2sql model(based on the fr2sql model[11]) and some of the methods for preprocessing the input sentence is adopted from some of the other models like Bag of words,Stemming,Stopword Filtering etc.The In2sql model/framework is based on a MVC framework and it works well with queries with simple retrieval(for single,multiple,all columns along with aggregate functions),having where conditions(one ,multiple,having certain conditional operators) ,inner and natural joins ,Order by,Groupby,MultipleQueries.Using the between operator and detection of values do not work efficiently.

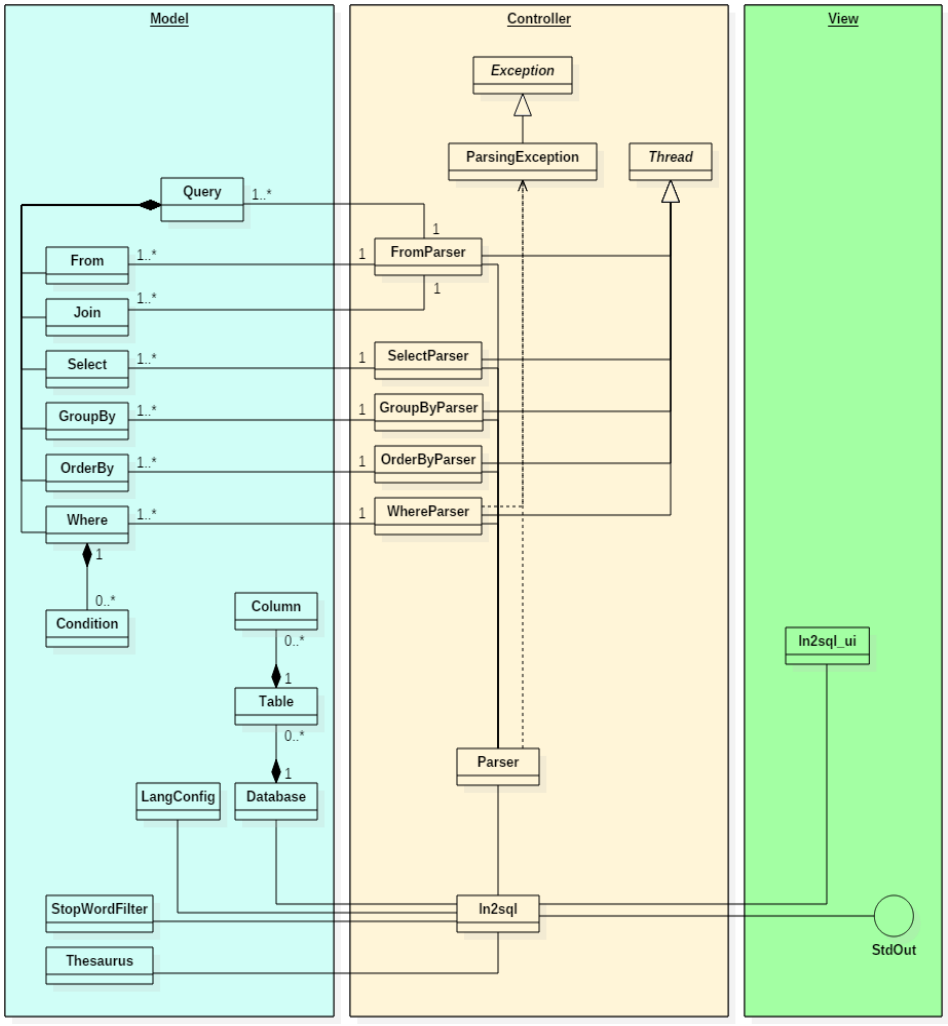
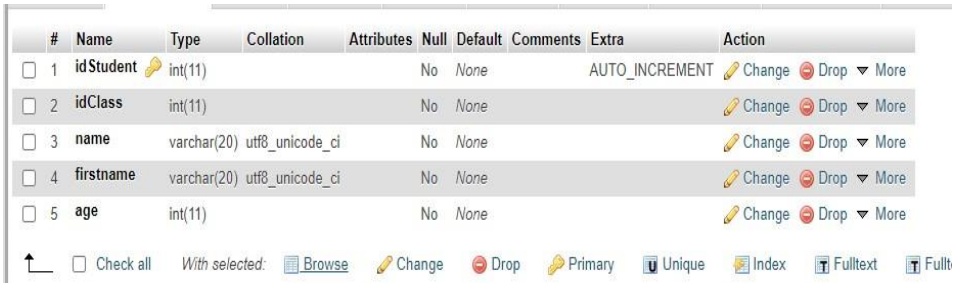


Figure 1.In2sql Framework

Implementation phase

Every SQL database will have a sql file associated with it. In order to query a database in natural language i.e, in simple words (and not following the commands according to the syntax) it is necessary to have a thesaurus file which contains all of the possible words available in the database and its synonyms. So to begin with the process we need an sql file, its corresponding thesaurus file , a file containing all of the stop words (words that do not have a significant meaning and will not be needed in the conversion process for eg: a ,the , am etc), and another file(can call it as conditional_keywords) which has the different conditional words along with their synonyms (equal ,greater than, less than ,and, between etc). The student table (Figure 3) is considered for further understanding of the process.



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	idStudent	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	idClass	int(11)			No	None			Change Drop More
3	name	varchar(20)	utf8_unicode_ci		No	None			Change Drop More
4	firstname	varchar(20)	utf8_unicode_ci		No	None			Change Drop More
5	age	int(11)			No	None			Change Drop More

Check all With selected: Browse Change Drop Primary Unique Index Fulltext Fullt

Figure 3. School Table used for testing the queries

The input sentence is taken from the user and the appropriate database and thesaurus files are selected. The conditional keywords, database, thesaurus and stop-words files are loaded. Basically a for each of the conditional keywords a list is created which contains its synonyms. In the same way a list is created for all of the stop words. The create and alter statements are considered from the sql file and all other commands like insert, update are ignored as we are concerned about formulating the query and not any sort of retrieval/update process. So we only need the details of the different tables and their columns along with the different constraints set on them. A dictionary is created for the loaded thesaurus file which contains the word and its synonyms. From this dictionary for each of the tables and columns their corresponding synonyms are loaded.

A parser object is created and its loaded with the database contents (table and column names along with synonyms) and the conditional_keywords contents. This parser object takes in the input sentence and the stop word list. The first step in processing the sentence is to check if the sentence has any of the stop words and remove them. So each word from the sentence is compared with the words in the list and removed if it exists. Finally we get the reduced sentence with the required words.

This sentence is now converted to a list or we can call it as a bag of words and if there are any special characters like , or !, they are neglected.We stem these words and convert any words that are in their plural formsinto singular noun format with the help of a LancasterStemmer available in the NLTK package in python. Now we try to determine all those words that will be part of the where clause. The basic process here is to ignore all those words occurring before a word which corresponds to a table name. So when a table name(or any synonym of the table name) is encountered all words before it along with the table name word are ignored and everything after it belongs to the where clause.

Generally column names before the where clause(in the select part)in an sql query will not be related to the where clause and hence if a column name occurs before the table name a variable is set with a value so that any other column names occurring after the table name will not be ignored and be considered in the where clause.

These set of words are converted back into a sentence by joining all of them with a space character and we try to extract all the possible value based words from it (For eg: age greater than 25,So 25 has to be extracted from it (Figure 4) . This is done by checking if the sentence has any conditional keywords and considering the values immediately after them. These conditional keywords are replaced with a combination of characters *res*@3#>>* (Figure 5) so that we can know that after this sequence there exists a value .

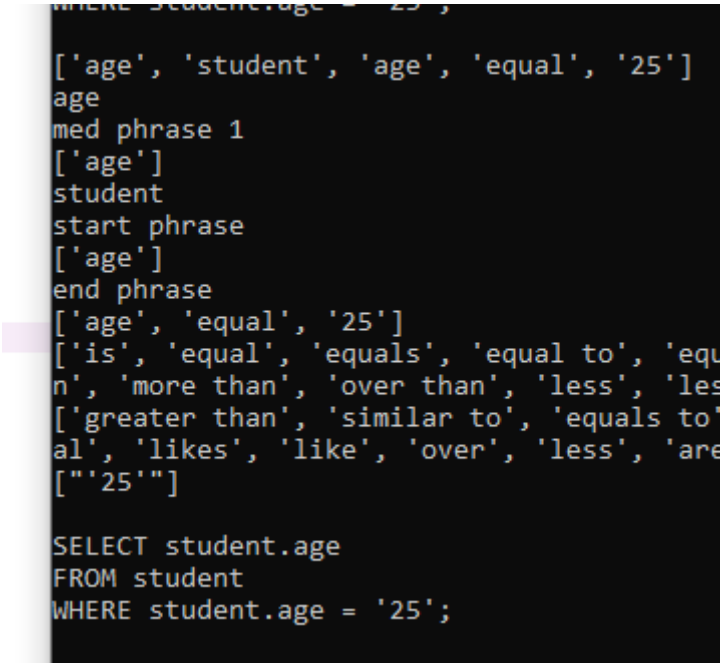


Figure 4. Extracting values for the where clause

Sometimes these values could be a combination of more than one word.Initially while splitting the sentence to cross-check with the list of stop words we have made sure that any value enclosed between “ ” are considered as a single word with the help of regular expression matching.So values with a combination of more than one word will be considered as a single word.Foreg: name equal to “PrincyAjit”.For such multi-word values, we try to replace the space character in the value with a combination of <_> characters to identify the entire sequence as one value. Later these characters are replaced back with spaces(Figure 5).

-

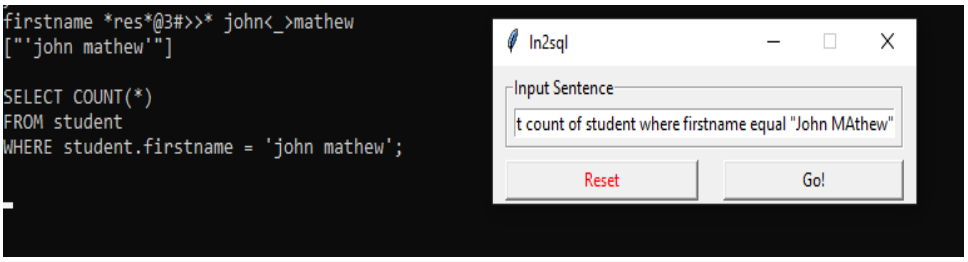


Figure 5. Replacing space character with <_> and operators with

res@3#>>*

The next step is to determine the words,table and column names from our input sentence that would belong to the select ,from and where clauses (Figure 6). For this we consider our input sentence once again as a list / token of words(a bag of words). Then we try to find the table and column names or their synonyms/equivalences in the list . When the first table name is encountered our select phrase will contain all the words until the table name. After which a variable number_of_table is set to 1 and this table is added to the tables belonging to the from clause(the other table names after this also get added to the from clause).

While checking for column names in the input statement firstly we check if this particular variable i.e, number_of_table is set to 0. If yes we consider the column name to belong to the select phrase else it belongs to the where phrase. Accordingly the from phrase will contain words from the list after the start phrase's words.And similarly the where phrase's word set will contain the words after the from phrase's words. Note that a new where clause will be constructed by merging the values from the previous where clause made above and the one we create now. This is done to ensure that any of the words have not been missed out. If no table or column names are found in the sentence an info is sent out that no keywords were found.

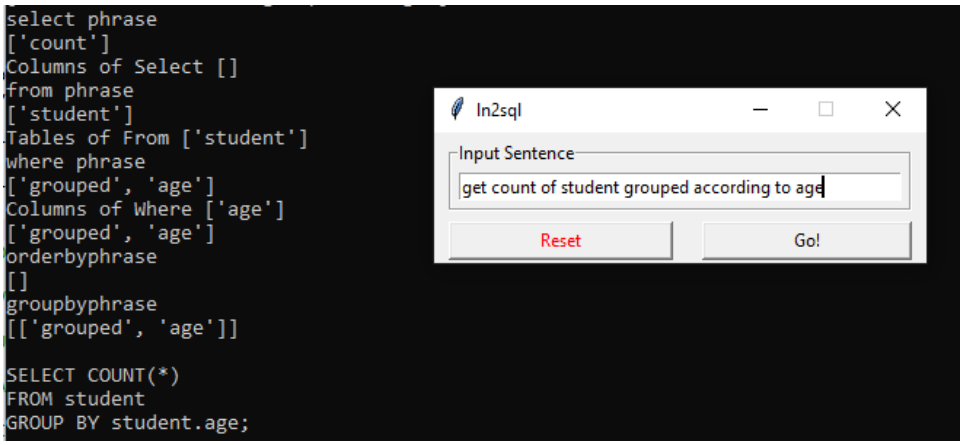


Figure 6 . Words belonging to different clauses/phrases for a groupby input

After this the from and where clauses which contain a set of words are cross- checked to find if there are any junction/disjunction i.e, and/or words or their synonyms and a count of these words is maintained. Then the where clause is checked for any keywords that could possibly mean either a group by/order by statement(Figure 6,7).

Accordingly those keywords are added into the group by /order by phrases along with their previous and following words(to know the column name the condition should be applied to and any other parameters like ascending/descending for order by) and if they do exist the where clause gets ignored(the new where clause will be empty if all the words in the old where clause implied either a order by/group by clause) (Figure 7).

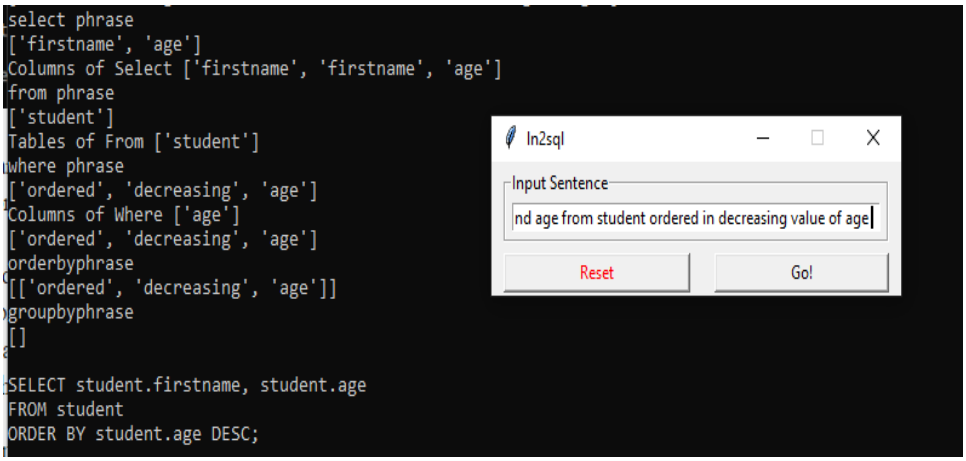


Figure 7. Words belonging to different clauses/phrases for an orderby input

Each of the words in these phrases will help in constructing the query as a whole. The select phrase's words are scanned to check if they have any of the conditional words(or their synonyms) like average,count,max,min,sum,distinct and accordingly the right keyword replaces it(According to the SQL syntax)(Figure 8,9).The corresponding column name follows the keyword. The from phrase basically contains the table name from which the content is to be retrieved. Similar to the select phrase ,the words in the where phrase are also scanned and the conditional keywords or their synonyms are replaced with an appropriate keyword.

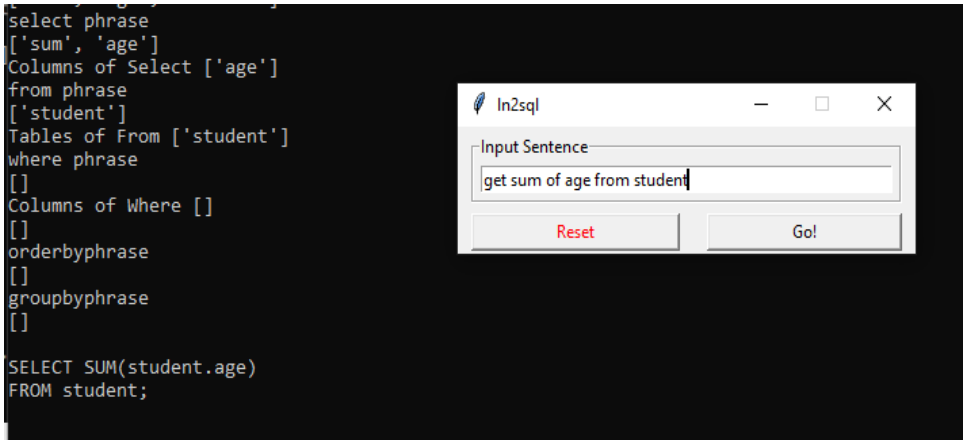


Figure 8. Query requiring the sum values

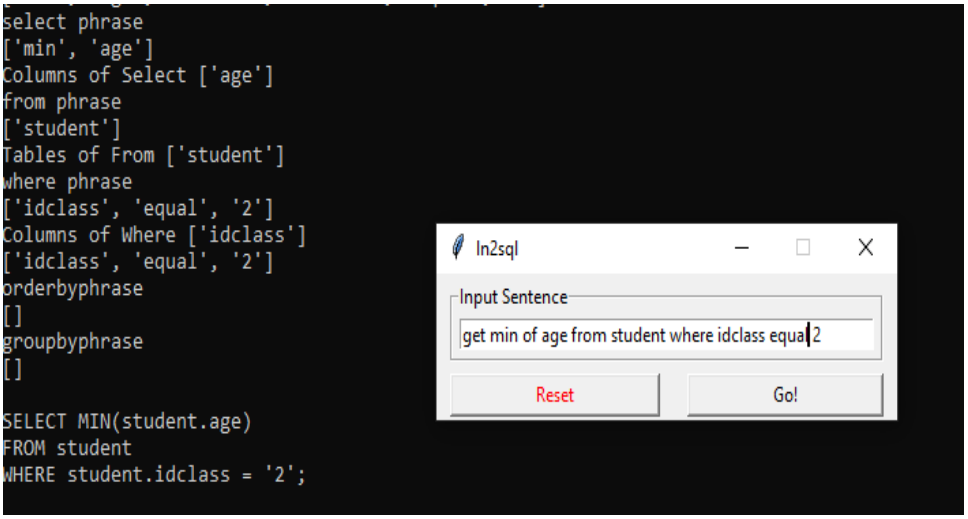


Figure 9. Query requiring the minimum values

Finally each of these phrases are joined together to give out our final query which could possibly be the solution that the user is looking for.

Conclusion

Here the main focus is formulating sql queries(select statements for retrieval) for simple retrieval(select * from table), with where conditions,orderby and groupbyconditions.The problem of having multi worded values was solved to an extent by having the words enclosed in “ ”. And then if these quotes were found the content within them was considered to be one word. The between condition is not able to detect its values and hence is not completely efficient(Figure 10).Similarly having complex words in the input might also give wrong results.The main intent was to understand the process on how words in natural language can be processed and converted into something useful.

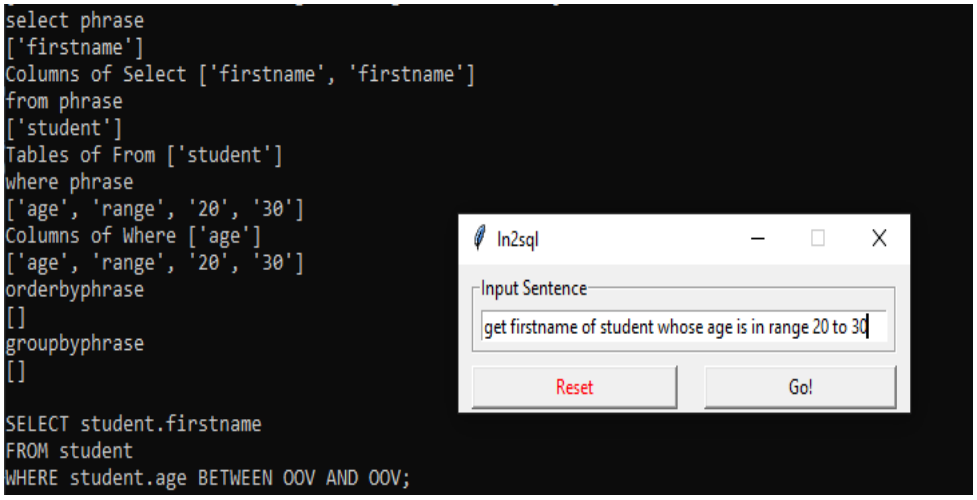


Figure 10. Between values not being identified

The process could be further enhanced by incorporating joins and working on the shortcomings such as having an efficient between condition. Also more focus should be on how to solve the problem of multi worded conditions and understanding the context of the sentence.By using efficient language based models like BERT and incorporating it here would make the process even more effective.

References

Garima Singh, Arun Solanki," An algorithm to transform natural language into SQL queries for relational databases", Selforganizology,3(3): 100-116,2016.
PrasunKanti Ghosh, SaparjaDey, SubhabrataSengupta, "Automatic SQL Query Formation from Natural Language Query",International Journal of Computer Applications.
Abhilasha Kate Satish KambleAishwaryaBodkheMrunal Joshi," Conversion of Natural Language Query to SQL Query", Proceedings of the 2nd International Conference on Electronics, Communication and Aerospace Technology (ICECA 2018).
[4]K. JavubarSathick and A. Jaya,"Natural Language to SQL Generation for Semantic Knowledge Extraction in Social Web Sources",Middle -East Journal of Scientific Research 22 (3): 375-384, 2014.
[5].Amit Pagrut, IshantPakmode, ShambhooKariya, VibhavariKamble
and YashodharaHaribhakta,"Automated SQL query generator by understanding a natural language statement",International Journal on Natural Language Computing (IJNLC) Vol.7, No.3, June 2018.
[6].Victor Zhong, CaimingXiong, Richard Socher," SEQ2SQL: Generating structured queries from natural language using Reinforcement Learning",arXiv:1709.00103v7 [cs.CL],9 Nov 2017.

[7].XiaojunXu,Chang Liu, Dawn Song,"SQLNet: Generating structured queries from natural language without Reinforcement Learning",arXiv:1711.04436v1 [cs.CL],13 Nov 2017.
UrsinBrunner,KurtStockinger,"ValueNet: A Neural Text-to-SQL Architecture Incorporating Values",arXiv:2006.00888v1 [cs.DB],29 May 2020.
PranaliNagare, SmitaIndhe, DhanashriSabale," Automatic SQL Query Formation from Natural Language Query", International Research Journal of Engineering and Technology (IRJET), Volume: 04 Issue: 03 | Mar -2017. [10]Xiyou Zhou, ZhiyuChen,"Translating SQL to Natural Language Queries with Pointer Mixture Model".
[11]BenoîtCouderc and Jérémy Ferrero."fr2sql : Database Query in French. (fr2sql : Interrogation de bases de donnéesenfrançais [in French])". In Proceedings of the 17th RECITAL (affiliated with the 22th TALN Conference). June 2015. Caen, France. ATALA. pp.1 -12.