Recent Trends in Intensive Computing
M. Rajesh et al. (Eds.)

doi:10.3233/APC210251

Natural Language Querying and Visualization System

Trishali Banerjee [a,1], Upasana Bhattacharjee [a] and Mrs. K. R. Jansi [a]
[a]*Student, Dept. of CSE, SRM Institute of Science and Technology, Chennai, Tamil Nadu, India*

Abstract. Data is the new gold; everything is data driven. But it is impossible for everyone to possess technical skills to be able to write queries and know different python tools used for data visualizations. The process of extracting information from a database is a mammoth task for non-technical users as it requires one to have extensive knowledge of DBMS language. But these data and visualizations are required for various everyday presentations and interactions in the professional world. This application would enable the users to overcome these obstacles. Our project aims at integrating two systems, an NLP interface to fetch data from simple English queries, and a second system where the fetched data with the help of natural language processing is used to form visualizations as demanded by the users will be created. This system would essentially help the people who are not techno-savvy or are not in the field of tech to interact with data using simple English.

Keywords. Natural Language Processing; Database; Python; SQL; Visualization

1. Introduction

Natural Language Processing basically helps us talk to a machine using the languages that are easily interpreted by human beings. It's an artificially intelligent technology. Imagine if a database were a person with whom you could actually talk to. Yes, NLP makes that possible. In recent times as the world has made a huge progress in this field, where processing huge amounts of textual information is done with an acceptable amount of efficacy. If we have to take an example, consider Web Search Engines, where these techniques form an essential component and thus served as an inspiration for us to make this querying and visualization system. Our application will provide the user with an interface to do just that. By using simple English sentences, interaction with the database and generation of queries will become possible. NLP integrated with Python will also help us to visualize data.
Visualizations will be created using simple English sentences. We have harnessed the immense capabilities of the python language as it is proven to be the best language for Machine Learning and artificial intelligence. A parsing model TAPAS which is out sourced by google has been used in training the machine, it extends BERT architecture

[1] Trishali Banerjee, Dept. of CSE, SRM Institute of Science and Technology, Chennai
Email. trishalibanerjee44@gmail.com.

which develops the Neural Network required for the purpose. The second approach is using Spacy which is again an open sourced module by python.

2. Related Work

Author Garima Singh and Arun Solanki in their paper "An algorithm to transform natural language into SQL queries for relational databases" have talked about ambiguity removal, that is a very important thing that they have addressed, the removal of repetitive attributes and clarifying with the user in an interactive way improves the accuracy and in that way the dependency on the machine also reduces and the flexibility increases. [11]. Santosh Waghmode and his students in their paper "SQL Query Formation for Database System using NLP", have used a unique Multinomial Logistic Regression Algorithm, this algorithm aims at predicting the type of query from the information that the user provides in the previous steps and they have also mentioned about the regular dictionary update that helps in achieving maximum accuracy.[10] For this project, the dictionary will be updated regularly not only for accuracy but also to address ambiguity. But the drawback of this system is that there exist a lot of complex queries that they have not addressed yet. Swapnil Kanhe, Pramod Bodke, Vaibhav Udawant and Akshay Chikhale in their paper "SQL Generation and PL/SQL Execution from Natural Language Processing", have mentioned their software will be able to generate both PL/SQL queries. In their algorithm they mention about a "Morphological Analysis"- Morphology is the structure of word. It is also concerned with derivation of new words from existing ones, Ex. Lighthouse (formed from light house). [2] They have also made sure the tables in the database are normalized for more accuracy. They have used c# .net frame work for the development of this application. George Obaido, Abejide Ade-Ibijola and Hima Vadapalli in their paper "Synthesis of SQL Queries from Narrations", have addressed the fact that SQL is not an easy language to master and their system would help the non- technical end user to interact with the RDBMS. They have created a color coded table and to convert NLP to query entities with matching color is found. They have also performed a 2 fold evaluation of the tool. First was using the crowdsourced XNorthwind dataset and second was using human subjects. [9]

3. Querying and Visualization System and Architecture

The natural language querying and visualization system is aimed at fetching answers to the queries then using that to visualize the data. The user inputs queries as simple English sentences, techniques such as matching, parsing, and tokenizing and dictionary mapping are utilized to generate the relevant query. This system will help in solving moderately difficult queries and as the system keeps learning the accuracy increases accordingly. The previous systems were based on forming queries and then fetching data. The formation of logical queries is difficult and according to the previously available data the accuracy is not satisfactory. Here the predictions based on denotations are done by selecting corresponding cells that are formed into vectors and also perform aggregations based on the user requirements. Also added is the visualization system, to give more meaning to the data and to give user an end to end

experience. An additional feature to this project being the use of Google Collab and Google Cloud for implementation.

In Figure 1, we can see the user gives input to the machine through the user interface. The data in the csv format is converted into data frame using pandas. Then the NLP pre-processing begins. It starts with breaking the input into individual words or sometimes phrases. These are then associated with a token id. Lemmatisation and stemming leads to the root word and clubs the words eliminating the suffixes in the word like 'ed 'or 's' or 'ing'. This helps reduce confusion. Then comes removal of stop words. Then tagging begins. Here basically the grammatical meaning is attached to the input. The tokens generated are tagged as nouns, pronouns, verbs etc. The data frame post this is converted into list of lists, this basically allows the tagging of user input to the data set to be easier and faster. Here the approach is different, the algorithm used is recurrent neural network and the modules used are tapas module open sourced by google and spacy which is also open sourced. The table is queried according to the meaning that has been attached to the user input and it is mapped to the required subset / value in the data set. In the result file the rows and columns are enumerated.

For the visualization again the data frame is formed, the user inputs the requirements and this is then passed through the matcher and we obtain the visualization appended in the GUI itself.

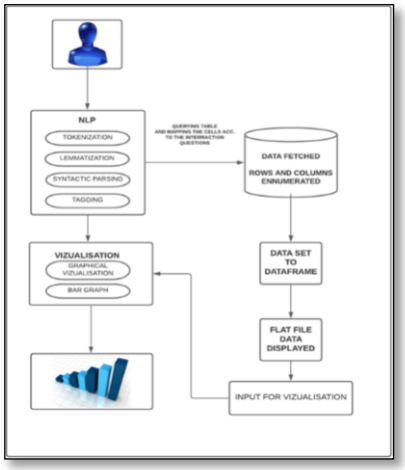

Figure 1. System Architecture

4. Modules

4.1. User input and Natural Language Query Pre-processing.

This module deals with the creation of an interface where the users will be able to give their inputs. Anyone who doesn't know how to write SQL queries can use this interface for the purpose of interacting with the database, the input as already mentioned will be a simple English sentence. It is very important to note that for NL processing everything has to be converted into string. Tkinter is used for the creation of graphical

user interface. The system first starts with pre-processing the natural language query entered, by carrying out the following.

Tokenization. The interface accepts input from the user in form of English sentences. The NLP system breaks the raw text input into small portions and generally separates each word. Here NLTK library is used to perform tokenization. Spacy Tokenizer has been used which is a modern technique for tokenization; much faster and easier, the advantage of using this method is one can specify special tokens that need not be segmented as well as those that typically have to be specified. Basically the tokens that one specifies takes precedence. There are many challenges during tokenization like there are way too many word boundaries available in the spread of English language.

Lemmatization. Lemmatization and stemming prune a word to its root. Let's say there is a phrase, "Display the names of managers who work in Bangalore" here, "names" becomes "name". Basically the words are converted to their base or root words. It is done to remove the endings of inflected words to return the base form of the word known as lemma. Lemmatisation and stemming contribute to reducing the number of common words (same words in different forms) and these in turn help amplify the accuracy of the tagging process.

Removal of Escape Words or stop words. Escape words are those words that are unnecessary for query generation and can be overlooked. Here is a list of few possible escape words that have been defined (tf_examples_utils is imported from tapas repository, it contains the stop words) A, An, The, Select, Find, Which, whose, Is, Of, A, With, To, for, Are, And, What.

Removing Ambiguous Attributes. Suppose there are 2 sentences, "display the students in the blue house" and "select the students who have the blue house as their reference location". In these 2 sentences "blue house" creates an ambiguity, in these cases an error alert is given out if the dictionary being used is unable to handle this.

Tagging. It basically pertains to converting a sentence to list of words or tuples in the form of "word, tag". It assigns grammatical information of a word. In simple words its parts of speech tagging, it tell us what part of the speech does the word belong to. Post tokenisation the tokens are classified into nouns, pronouns, verb or the type of variable mentioned. double, integer, string. Then the complete thing is passed through an IDENTIFIER where the generated tokens are classified into relations, attributes and clauses.

Dependency Parsing. It is a process of analyzing the grammatical structure or the grammatical construct of the sentence and also analysing what are the dependencies between the words in the sentence. There are various tags. .

Named-entity recognition. NER also called entity identification. It is a subset or sub task.

Name Stream. Proper nouns are crucial for query generation and visualization. Spotting them in English is also easy since they start with a capital letter, also if they comprise of a group of words the first letter is always capitalized such group of words are not broken down and treated as a single token. We also save these words token wise (dual approach) that helps us with the ambiguity just in case if any of the word is missing or if it is the same thing reframed in a different way.

4.2. Formation of a data frame

The first important step is to convert the data set into python data frame. This will allow the modules and python function to work on the data frame. The flow of the software is after the data is fetched post tagging, it is either displayed like a flat file or the user demands for visualization. Data is first converted into a data frame. From here there are two possibilities either using tensorboard for creating visualization or matplotlib. Here Pandas is used for the formation of the data frames. The original data is in the .csv file format that is converted into a data frame with rows and columns. The query entered by the user is tagged to these tuples. The data frame is converted into a "list of lists." That way the tagging becomes way easier and also increases accuracy.

4.3. Fetch data on the basis of the Natural Query

This system, as already mentioned uses TAPAS module and imports tensorflow from the python libraries (there are many imports but they are quite generic). Tensorflow incorporates the deep learning algorithm- **Recurrent Neural Network and to be more specific Long Short- Term Networks (LTSMs).**This helps in mapping the complete NL query or as the system calls it an "Interaction Question". Pre-processing the question is marked with an id. Essentially each cell in this algorithm is converted into a vector and the algorithm according to the user input maps it to the particular cell "answer coordinates" it wants to point towards. Also used is an "annotator", it keeps track of the question and table pair. This helps in case when questions are repeated or rephrased as something else but mean the same. There are 2 approaches that is used to fetch data, the first one is discussed above (tapas module). The second software library used is Spacy.

4.4. Input about the visualizations

Regardless of the imports being used, deciding the requirements of user-input for the visualisations is slightly difficult as each plot is built separately and it is a work in progress. The user requirements are tagged and the interface is updated accordingly. Seaborne and Matplotlib are the 2 python modules used for the visualisation and tagging to get the input to match the visualisation requirements.

5. Algorithm

Tapas uses a Bert file for the creation of the neural network. The Artificial Neural Network or the ANN works identical to a human brain. Specifically the software uses Long Short Term Memory Network Algorithm. NLP pre-processing is triggered using the artificial neural networks. There are different layers to the network generated. There are multiple inputs given to the system. In this case a stream of words and the output obtained is a single output, here the "query". The strategic reasoning is done with the help of a radial basis function. An algorithm which remembers the order of dependencies for the sequence of input, this helps in tagging the terms with accurate weight and in the long run helps in ranking of terms in accordance of their importance. LTSM it can choose to modify and remember or forget a piece of information selectively which makes sequential processing easier. LTSM works on a Cell State mechanism and there are 3 states to it. First, the previous cell state which describes the

information that was present in the cell in the previously timed step. Second, the previous hidden state describes the output of the hidden state previously. Third, the current input. The input is passed through the different layers of the Network. The network used is Feedback network because feed forward network thinks in one direction only and never looks back, that is an important functionality of the algorithm to be used because we are looking at a Sequential stream of data. There are 2 parts in an RNN first is an encoder it is responsible for encoding symbol sequences to fixed length representation in vector format. The data set, the list of list that was generated is converted into vector format ready to be tagged and queried. The decoder does the opposite, brings down the symbol to be displayed or used further for visualisation. In addition to that Matcher. Match is used in spacy .Here the user inputs an argument and a group of possible cases to match with. There are 4 types of cases available but we have used "values". Values because the argument is matched against a list of values specified. Spacy uses Deep Neural Network that is based out of Convolution Neural network. It is built over a frame work Embed, encode, attend and finally predict. Bloom filter and shift reduce parser is used, the hash codes of the words are generated and kept in the dictionary, this helps in maintaining a compact dictionary, the collision is also less hence ambiguity removal. Then DNN is used to encode list of words to a matrix of sentences and ids are generated. Attend is removal of stop-words and maintaining only the parts which contribute to the meaning. Processing through the remaining layers and tagging results are generated.

6. Results

The data is being fetched successfully, and is accurate to the best of knowledge. Since, the machine is still in the learning phase, to reach desired accuracy the dictionary in use has to be regularly updated.

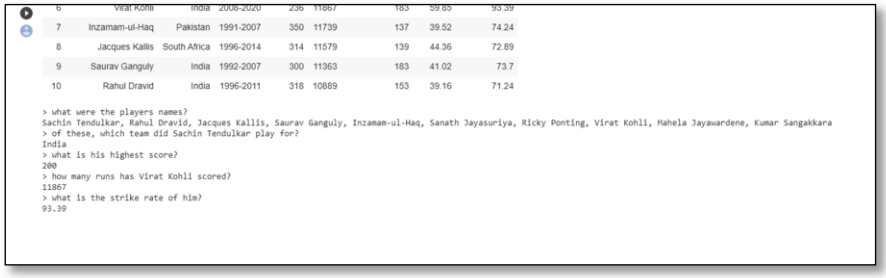

Figure 2. Output from Cricket Dataset

Figure 2 gives an idea of the output being generated. The system currently can seamlessly produce bar graphs to provide the users with a visualization of the data. Figure 2 shows the output carried out on a Cricket data set. It can be seen here that the user queries the database with variety of questions .It can also be seen here that the system is capable of answering progressive questions involving pronouns, like, "What is his highest score?" etc.

```
    29              0.41            0.46            2

> maximum satisfaction level where salary is low
0.84, 0.78
```

Figure 3. Output from Employees dataset.

Figure 3 shows the output from an Employee dataset

```
question_id id  annotator position  answer_coordinates  answers
0-0_0 0 0 0 ['(12, 1)', '(17, 1)']
"[{""column_index"": 1, ""row_index"": 12, ""begin_token_index"": 0,
 ""end_token_index"": 3, ""token_ids"": [1014, 1012, 6391], ""score"": 0.9626243710517883},
  {""column_index"": 1, ""row_index"": 17, ""begin_token_index"": 0, ""end_token_index"": 3,
   ""token_ids"": [1014, 1012, 6275], ""score"": 0.6118265986442566}]"

```

Figure 4. Generation of question ID.

Figure 4 shows how the Question id is generated. The role of generating a question id is the next time a similar question is asked by the user, the id is fetched and the results are returned in a shorter period of time. Hence, question id helps make the process more efficient.

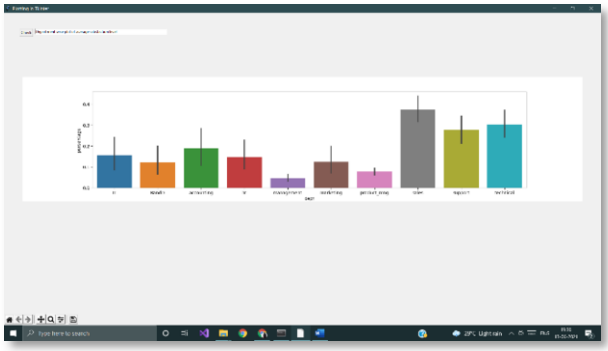

Figure 5. Visualization Output.

Figure 5 shows the visualization that is generated on the Employee dataset to satisfy the user's requirements.

7. Conclusion

NLP is one of the few tools which can change the complete working of the computer program interface. This system can not only query the database and return the required results to the user but also provide the user with the option to visualize their data in a way that will make the results of their query even clearer. Dealing with ambiguity removal and constant update of the dictionary being used helps improve the accuracy of the system. Currently, this system is capable of producing bar graphs to visualize the data. Since all forms of visualization have separate requirements, the future scope of the project is to include varieties of visualizations and also refining the library of querie

References

[1] Dey, S., Gosh, P.K. and Sengupta, S. Automatic SQL Query Formation from Natural Language Query. International Journal of Computer Applications (0975-8887) & International Conference on Microelectronics, Circuits and Systems (MICRO-2014).

[2] Bodke, P., Chikhale, A., Kanhe, S. and Udawant, V. SQL Generation and PL/SQL Execution from Natural Language Processing. International Jornal of Engineering Research & Technology (IJERT) ISSN. 2278- 0181, Vol. 4 Issue 02, February-2015. URL. www.ijert.org.

[3] Nguyen, D.Q.[Dat], Nguyen, D.Q.[Dai] and Pharm, S.B. Ripple Down Rules for Question Answering. Semantic Web 0 (2015) 1-0 IOS Press.

[4] Goswami, A., Gupta, P., Koul, S. and Sartape, K. IQS-Intelligent Querying System using Natural Language Processing. International Conference on Electronics, Communication and Aerospace Technology ICECA 2017.

[5] Srinivasan, A. and Stasko, J. Natural Language Interfaces for Data Analysis with Visualization. Considering What Has and Could Be Asked. Eurographics Conference on Visualization (EuroVis) 2017.

[6] Patil, S.J. Python- Using Database and SQL. International Journal of Science and Research (IJSR) ISSN. 2319-7064 Impact Factor (2018). 7.426.

[7] Haribhakta, Y., Kamle, V., Kariya, S., Pagrut, A. and Paknode, I. Automated Sql Query Generator By Understanding A Natural Language Statement. International Journal on Natural Language Computing (IJNLC) Vol. 7, No. 3, June 2018.

[8] Kombade, C., More, M., Pujari, A. and Patil, S. Natural Language Processing with some abbreviation to SQL. International Journal for Research in Applied Science & Engineering Technology (IJRASET) ISSN. 2321- 9653; IC Value. 45.98; SJ Impact Factor. 7.177. Volume 7 Issue XII, Dec 2019- Available at www.ijraset.com.

[9] Potharaju, S. P., & Sreedevi, M. (2017). A Novel Clustering Based Candidate Feature Selection Framework Using Correlation Coefficient for Improving Classification Performance. Journal of Engineering Science & Technology Review, 10(6).