2021 6th International Conference for Convergence in Technology (I2CT)
Pune, India. Apr 02-04, 2021

Natural Language Processing Technique for Generation of SQL Queries Dynamically

Hrithik Sanyal
Department of Electronics & Telecommunications
Bharati Vidyapeeth Deemed University
Pune, India
hrithiksanyal14@gmail.com

Sagar Shukla
Department of Electronics & Communications
Gyan Ganga Institute of Technology
Jabalpur, India
sagarshukla840@gmail.com

Rajneesh Agrawal
Mentor,
Comp-Tel Consultancy
Jabalpur, India
rajneeshag@gmail.com

Abstract— **Natural Language Processing is being used in every field of human to machine interaction. Database queries although have a confined set of instructions, but still found to be complex and dedicated human resources are required to write, test, optimize and execute structured query language statements. This makes it difficult, time-consuming and many a time inaccurate too. Such difficulties can be overcome if the queries are formed dynamically with standard procedures. In this work, parsing, lexical analysis, synonym detection and formation processes of the natural language processing are being proposed to be used for dynamically generating SQL queries and optimization of them for fast processing with high accuracy. NLP parsing of the user inputted text for retrieving, creation and insertion of data are being proposed to be created dynamically from English text inputs. This will help users of the system to generate reports from the data as per the requirement without the complexities of SQL. The proposed system will not only generate queries dynamically but will also provide high accuracy and performance.**

Keywords—Natural Language Processing, Parsing, Lexical Analysis, Structured Query Language, English Text

I. INTRODUCTION

Natural Language Processing (NLP) being a sub-branch of Artificial Intelligence is mostly used in the fields where and when we need a human to machine bonding or communication. It is required when we have high loads and humans are to provide the inputs as per the requirements by the system. Without these techniques, it would be very difficult, because of its low performance and involving errors and mistakes by a human. Thus, NLP has become very promising in solving such errors and challenges and now considered the best method. Not restricted to texts alone, it is being widely used in multifarious applications such as audio, video, animations, etc.

Natural Language Processing has many wide numbers of algorithms. These algorithms are specific to the inputted languages and are commonly applied to most real-world languages.

Natural Language Processing also written as NLP, helps machines and computers to interact with humans using languages as the input. The ultimate foal of NLP remains the very same, for reading, then deciphering, next understanding and finally make sent. Making sense of languages inputted by humans is very important. It has to be differentiated what is valuable and thus needs to be processed and work on, and what is not.

In section 1, introductions of Natural Language Processing have been discussed. In section 2 explanation of Natural Language Processing techniques is elaborated. In section 3, the Parsing technique is discussed and in section 4, Structure Query Language is expounded. In section 5, existing systems have been discussed. Section 6, describes the proposed work and section 7 elaborates the expected outcome. Section 8 shows conclusions and future scope is discussed.

II. NATURAL LANGUAGE PROCESSING METHADOLOGY

NLP techniques mostly rely on Machine Learning or Deep learning for successfully implementing a bond between machines and humans. Some of the methodologies are as follows:

1. Named Entity Recognition (NER)
2. Tokenization
3. Stemming and Lemmatization
4. Bag of Words
5. Natural language generation
6. Sentiment Analysis
7. Sentence Segmentation

A. Named Entity Recognition (NER)

Named Entity Recognition is not only a very popular technique but also it has many merits. Used often in Semantic analysis which means, the meaning, be it hidden or not, conveyed by any text. In this method, algorithms take inputs as paragraphs or sentences or phrases, and then quickly identifies all the names or nouns present in them. Some examples are as follows:

News Categorization: The main purpose of this algorithm is to scan news articles and then understand and finally extract important pieces of information, be it of any kind, like, companies, celebrities, health-related, scientific discoveries, politics, individual people, etc. It is most helpful for categorising information into different categories.

Efficient Search Engine: This algorithm helps to extract information from any article and marking them with relevant tags and thus storing them separately

978-1-7281-8876-8/21/$31.00 ©2021 IEEE

according to their categories. This will help boost searching by users.

B. Tokenization

The term Tokenization means to split a whole phrase or text into a different number of lists of tokens. Small tokens can be anything, be it words, small phrases, numbers, punctuations, characters, etc. The main merits of Tokenization are:

1. Reduce search timings, thus making them fast and efficient.
2. Since reducing unnecessary searches, hence saving a lot of memory or storage.

Talking more on Tokenization, it may be considered as the sub part of Information Retrieval (IR) system. This means that it will not only process the text, or phrase but also generate the tokens alongside. This may be useful for indexing purposes. Amongst many algorithms techniques available, the most efficient algorithm is Porter's Algorithm.

The most basic working of NLP is to convert phrases or strings to words, which are again converted further to characters. Albeit this sounds very easy and convenient, but actually is not. The main issue with NLP is that it is mostly not able to understand or comprehend the hidden meaning behind the text

C. Stemming and Lemmatization

Stemming is defined as the process to reduce the derived words from their base or root word form. This is basically done by cutting off the suffixes. For Example, applying stemming on words like "Eating" or "told" will make them "Eat" and "tell".

Similarly, Lemmatization basically deals with the best use of vocabulary or morphological analysis of words. This is done to remove abrupt endings and give a proper dictionary form of any character or word, which is indeed known as "Lemma". It basically removes unnecessary words by interpreting the (POS) Part of Speech or context of the word.

D. Bag of Words

Bag of words techniques uses Machine Learning modelling. It is used to process the text beforehand and then the extraction of all the features is done. The main reason behind it is referred to as "Bag" is due to its mechanism which concerns the times a particular word is being repeated. It is not at all concerned with its locations.

E. Natural Language Generation

Natural language generation (NLG) can be defined as a technique which uses structured data (which is raw) and then converts them into a plain language, such as English, Hindi, Bengali, etc. Also commonly referred to as "Data Storytelling", this techniques is helpful in organizations which deal with a prodigious amount of data. Using this technique, raw data is converted to natural language which is then understood better by people. Often seen as the very stark opposite of NLP. Unlike NLP, NLG converts raw data to an interpretable text which is very much understandable, for all the people.

F. Sentiment Analysis

When we talk about Natural Language Processing, the most common technique that comes to mind is the Semantic Analysis. This is by far one of the most helpful techniques for successfully implementing NLP. By the help of Semantic Analysis, we may interpret or comprehend the underlying emotions or feelings of any text. Also referred to as "Emotion Artificial Intelligence" or "Opinion Mining". The functionality of it is finding the emotion behind any text. The emotions may be positive, neutral or negative. So Semantic Analysis is basically used to find whether the text is sounding positive or negative or neutral. Also referred to as "Finding The Polarity Of Text". For better results, subjective data should be used which are basically statements or facts that contains no feelings or emotions rather than the objective data which contains human feelings or emotions.

G. Sentence Segmentation

Sentence Segmentation helps in dividing the text into meaningful and comprehensive phrases. It involves, identification of sentences among words in texts. This is done by the use and help of punctuations found in the texts.

Also called as sentence boundary detection, sentence boundary disambiguation or sentence boundary recognition. NLTK, Stanford CoreNLP, Spacy are some of the libraries that are freely and easily available for the users.

III. Parsing

The term "Parsing" finds its roots in the Latin word 'pars' (which means 'part'). Parsing can be defined as a technique for the extraction of meaning or emotion from any text. Commonly referred to as Syntactic analysis or syntax analysis.

The main idea is to check for the meaningfulness of any text.

To give it a proper formal definition, one can define it as the process involving analysis of phrases contain symbols in natural language, along with going with the rules of grammar as well. There are many types of Parsers available. Some are listed below:

A. Recursive descent parser

Considered one of the most practical types of parsing. Following a top-down process, it tries to verify the syntax of the inputted data, whether it is right or not by reading characters from the input and then matching them with grammars. To do this, it tries to read the inputted data from left to right.

B. *Shift-reduce parser*

This type of parser follows a bottom-up method. It tries to search a sequence of the word until and unless the whole phrase gets reduced. To do so, it creates a parser tree to the start symbol.

C. *Chart parser*

This type of parser is mostly used when the grammar or phrase is ambiguous. It makes use of dynamic programming for solving parsing issues, due to which it stores results in structures called "charts". It is re-usable.

D. *Regexp parser*

Regexp parsing is one of the most used parsing technique. It makes use of regular expressions for parsing input sentences and thus generating a parse tree from it. The regular expression is defined in the form of grammar on top of a POS-tagged string.

E. *Deep Vs Shallow Parsing*

Deep Parsing is basically suitable and used for complex applications related to NLP. In deep parsing, the searching tries to give a syntactic structure to a phrase. Some of the best instances, involving deep parsing are Dialogue System and Summarization. Also commonly referred to as "Full Parsing".

Unlike Deep Parsing, Shallow Parsing is suitable for less complex application related to NLP. It unlike deep parsing parses a limited and short part of the syntactic information from the given phrase. Some of the best instances are Text Mining and Information Extraction. Commonly referred to as "Chunking".

IV. Structure Query Language

Databases are the backbone of any system which keeps a large amount of data generated by the system. Databases not only keeps the data intact in tabular formats, but they also have special language, statement of which provides a mechanism to store, maintain, retrieve and manage the data in them. This specific language termed Structured Query Language (SQL) has been modelled and standardized by Dr E.F. Codd for databases and rules designed by him draws an acceptability line for the databases. Although SQL is written in simple English language the syntax and rules make it difficult and hence companies require a specific database team for handling data-oriented tasks and query writing. This poses an extra burden for the companies and causes a lot of dependencies.

Different operations involved in SQL viz SELECT, CREATE, UPDATE, and DELETE makes further involvement for the developers and the addition of procedural language enhances the requirement of skilled persons as part of the team. This work provides a cheaper yet firm, accurate and simplified system for the generation of queries with the help of natural language processing techniques.

V. Existing Systems

Automatic code generation has been enhancing the Software Engineering and Software Development projects. For a very long time, researches have been conducted for the advancing of code generations for Web-Based DBMS systems. A wide number of prototypes of tools have been developed for debugging the source-to-source codes related to c and c++ applications. NLP is also very useful when it comes to software code generation and its applications for Graph databases.

Graph Databases are widely popularizing Artificial Intelligence applications, etc, by using Cypher a query language that allows users for searching without actually knowing to program. Through this paper, the use of NLP is being proposed specifically for users and not familiar with programming skills. It shows 3 use-cases, translation of generic English phrases for OpenCypher and then specialized graph engines like Huawei EYWA [1].

In today's time, every electronic device is connected to the internet. Everybody around the world could retrieve information from anywhere anytime using the internet. This information is stored in the form of databases. But accessing the databases directly without any prior knowledge becomes daunting and challenging. Hence there should be a system that helps the users to access the information from the databases. With this paper, the main focus is on developing a system that would take questions as input and output SQL queries. This is achieved by using NLP, for accessing information related to railways reservation databases. This database contains a set of 2880 structured natural language queries on fare and seats available. For these steps like tokenization, lemmatization, parts of speech tagging, parsing and mapping were involved. Also, accuracy of 99=8.89% was achieved. An overall view of NLP usages and expression for mapping query in English languages into SQL is also present [2].

A very powerful tool for retrieving or even managing data stores in a DBMS is called Structured Query Language (SQL). But for this accessing of data, a very intellect and knowledge of SQL are required. For allowing users to use DBMS, natural language interfacing to databases is being developed which would use natural language queries as queries. This is called Natural language Interface to Database (NLIDB). For developing such a system, one has to bridge the gap between the underlying data and natural language query (NLQ). Keyword mapping a task which is used for mapping of individual keyword into the NLQ databases. It becomes very difficult to comprehend the true meaning of each word in NLQ and thus the whole process becomes challenging and daunting. Through this paper, a system called MyNLIDB is being developed that would have a good performance. It would use a Schema-Graph using an underlying database, Stanford part-of-speech parser and dependency parser using pipeline processing for converting NL Query to SQL. MyNLIDB is both domain and database independent. It works better for simple queries [3].

This work mainly focuses on the Question Answering (QA) problem which is also one of the most popular tasks of NLP. For this, an extensive process is semantic parsing, which uses synchronous frameworks for deriving semantic & syntactic structures of texts. The paper proposes an approach that aims in answering questions efficiently by semantic parsing of texts by using a method that is based on lambda calculus to derive logical sentences. Firstly, by collecting significant feature-sets the questions are analysed and correct answers are given. This work also focuses more on lambda calculus-based parsing, unlike traditional approaches. Through this proposed work, an accuracy of 95% is achieved for all the YES /NO questions. The overall accuracy is around 83% including the five tasks of bAbI- 10k question answering datasets [4].

Both quantitative and non-quantitative data have a great impact on the financial. With the help of this paper, a CNN model related to stock price prediction is being proposed. It is made using Korean Natural Language Processing. Korean sentences were converted first into nouns and later used to extract words. These words and sentences were used as input data for the CNN model and thus the prediction of the stock price was made after a training period of 5 days. Compared to other models, which have an accuracy of 50%, this model achieved an accuracy of 53%, which is also the highest indeed [5].

NLP has not only enhanced computers but also plays an important role in the development of artificial intelligence. Not only AI, but NLP is being used frequently for helping people survive. This has been possible by the question-and-answer system which is being described. Through this paper, we propose a system that would mainly work on natural language processing and other technology for information retrieval. Albeit, different from the traditional search engines, but is mainly based on text retrieval. The main working of traditional search engines is as it requires users to input a series of keyword combinations and which would suggest users multifarious websites related. This system may get not only accurate but short and small answers for the users, helping them in their day-to-day life. This paper illustrates the question answering system of NLP and how it is further optimized [6].

VI. Proposed Work

In this work, parsing, lexical analysis, synonym detection and formation processes of the natural language processing are being proposed to be used for dynamically generating SQL queries and optimization of them for fast processing with high accuracy. The complete work shall be formed in the following steps:

1. Taking user inputs through a simple interface
2. NLP Parsing for tokenization of the input text
3. Stemming & Stopping of the unnecessary words
4. Use of Logic to decide the type of query formation (CRUD – Create, Retrieve, Update/Insert, Delete)
5. Query formation
6. Query Optimization
7. Query Execution & Result generation and providing the same in the form of reports to users
8. Evaluation of Time required and verification of results from the already existing queries created for testing

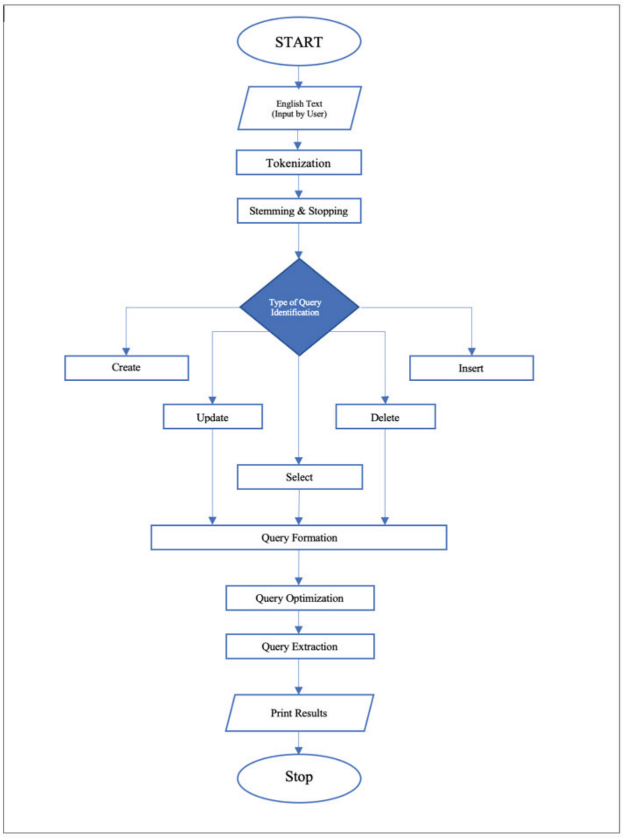

Fig. 1. Methodology of Natural Language Processing

VII. Expected Outcome

NLP has been used in the processing of the different human languages and generating results from them. Application of NLP in SQL Query generation has been promising in past and in this work, we have focused on the high performance of the SQL query generation. The proposed system as above has been implemented using the NLP library of Python and leveraged the features such as Numpy Arrays, Tokenization, stop word removal, filtered word mechanism of NLP have been used along with the custom query words list for the formation of SQL query etc. The implementation has been executed with a set of English sentences and generation of SQL Queries and got the results tabulated and charted below.

TABLE I. RESULTS OBTAINED FROM DIFFERENT EXECUTIONS OF THE IMPLEMENTED SYSTEM

SNO	ENGLISH TEXT	QUERY FORMED	PROCESSING TIME (MS)	TOTAL PROCESSING TIME (MS)
1	get all states	select state from cities	2699.28	2876.18
2	get cities and states	select city, state from cities	2225.02	2324.96
3	fetch cities of Madhya Pradesh	select city from cities WHERE state in ('Madhya Pradesh')	10480.13	10538.09
4	fetch all cities of India	select city from cities	1773.23	1881.17

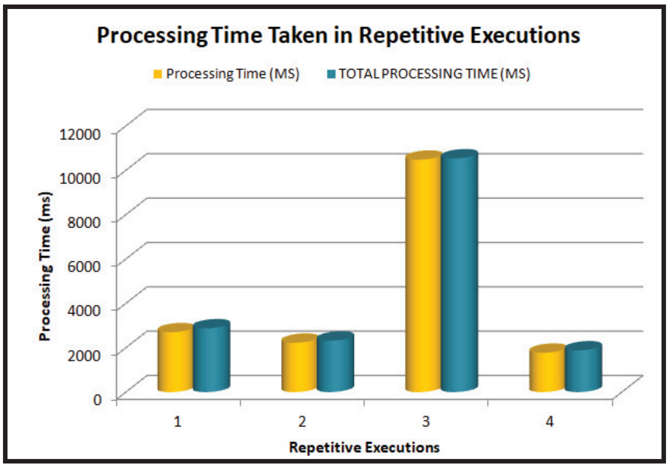

Fig. 2. Processing Time Taken in different executions of the implementation of the proposed system

Inference: The graph above has been created using the time taken in processing of the proposed implementation and the total time is taken from tokenization to result in display. From the graph, it is clear that the time taken in report generation is not much and hence the difference is small. Most of the time taken is in tokenization, stop word removal and filtering of the tokenized words. The actual time taken in SQL query formation is less as the algorithm implemented has constant time complexity resulting in negligible time contribution. Overall, the system is having high performance than the existing systems.

VIII. CONCLUSION & FUTURE SCOPE

This work expected that the system to generate queries dynamically and execute them against the standard database e.g. MySQL. The various steps have been implemented and measured for accuracy and performance to compare them with the existing database. The results expected from the proposed system i.e. generate accurate results for the users for different types of queries is working exactly as expected.

Although there are works related to NLP based query generation that has been done in the past by the researchers but studies show that most of them are merely surveys or limited only to retrieval queries. All of them have only discussed the success of the formation of SQL statements have not given considerations for the performance of their systems. This work has not only focused on the generation of dynamic SQL statements but also considering the performance and accuracy of the queries formed dynamically along with different types of queries formation. As expected, that the system is providing high-performance queries. As per the results obtained the system is having high performance and the results are accurate too.

The system can be tested in different databases in future for its accuracy and high performance. It can be made generalized by adding more and more custom lists for the formation of queries. More different words from human languages can be added to the test and make the system more flexible.

REFERENCES

[1] G. J. D. R. Hains, Y. Khmelevsky and T. Tachon, "From natural language to graph queries," 2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE), Edmonton, AB, Canada, 2019, pp. 1-4, doi: 10.1109/CCECE.2019.8861892.

[2] M. Uma, V. Sneha, G. Sneha, J. Bhuvana and B. Bharathi, "Formation of SQL from Natural Language Query using NLP," 2019 International Conference on Computational Intelligence in Data Science (ICCIDS), Chennai, India, 2019, pp. 1-5, doi: 10.1109/ICCIDS.2019.8862080.

[3] A. Das and R. C. Balabantaray, "MyNLIDB: A Natural Language Interface to Database," 2019 International Conference on Information Technology (ICIT), Bhubaneswar, India, 2019, pp. 234-238, doi: 10.1109/ICIT48102.2019.00048.

[4] J. Sarker, M. Billah and M. A. Mamun, "Textual Question Answering for Semantic Parsing in Natural Language Processing," 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), Dhaka, Bangladesh, 2019, pp. 1-5, doi: 10.1109/ICASERT.2019.8934734.

[5] H. Yun, G. Sim and J. Seok, "Stock Prices Prediction using the Title of Newspaper Articles with Korean Natural Language Processing," 2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), Okinawa, Japan, 2019, pp. 019-021, doi: 10.1109/ICAIIC.2019.8668996.

[6] K. Zhang, "Research on the Optimizing Method of Question Answering System in Natural Language Processing," 2019 International Conference on Virtual Reality and Intelligent Systems (ICVRIS), Jishou, China, 2019, pp. 251-254, doi: 10.1109/ICVRIS.2019.00069.

[7] H. Sanyal and R. Agrawal, "Study of Parsing Techniques for Natural Language Processing – A Comparison" International Research & Analysis Journal (IRAJ), India, ISSN (e): 2349-9788 Volume-14, Issue-1,Feb-2019.
doi: http://ira-journal.com/pdf/1412018/p14_1_8.pdf

[8] Sathick, K Javubar and Jaya, A, (2015) "Natural language to SQL generation for semantic knowledge extraction in social web sources," Indian Journal of Science and Technology, vol. 8, Issue 1, pp. 1–10.

[9] Singh, Garima and Solanki, Arun, (2016) "An algorithm to transform natural language into SQL queries for relational databases," Selforgani- zology, Directory of Open Access Journals, vol. 3, Issue 3, pp. 100–116.

[10] Huang, Bei-Bei, Zhang, Guigang et al., (2008) "A natural language database interface based on a probabilistic context free grammar," IEEE International workshop on Semantic Computing and Systems, pp. 155– 162.

[11] Rao, Gauri, Agarwal, Chanchal, Chaudhry, Snehal, et al., (2010) "Nat- ural language query processing using semantic grammar," International journal on computer science and engineering, vol. 2, Issue 2, pp. 219– 223.

[12] Satav, Akshay G, Ausekar, Archana B and Bihani, et al., (2014) "A Proposed Natural Language Query Processing System," International

Journal of Science and Applied Information Technology, vol. 3, Issue 2, pp. 219–223.

[13] Iftikhar, Anum, Iftikhar, Erum, Mehmood and Muhammad Khalid, (2016) “Domain specific query generation from natural language text,” IEEE Sixth International Conference on Innovative Computing Technol- ogy (INTECH), pp. 502–506.

[14] Pooja A Dhomne, Sheetal R Gajbhiye, Tejaswini S Warambhe, and Vaishali B Bhagat. Accessing database using nlp. *IJRET Int. J. Res. Eng. Technol. eISSN*, pages 2319–1163, 2013.

[15] R. Grmek, Y. Khmelevsky, and D. Syrotovsky. Automated inventory tracking system prototype in cloud. In *High Perfor- mance Computing and Simulation (HPCS), 2011 International Conference on High Performance Computing & Simulation*, pages 435–441, Istanbul, Turkey, July 4-8 2011. In Cooperation with the ACM, IEEE, IFIP, Co-Sponsored by IEEE Turkey, ASIM, EU- ROSIM, CASS, JSST, LSS, PTSK, TSS, Bahcesehir University.

[16] G. Hains, Chong Li, D. Atkinson, J. Redly, N. Wilkinson, and Y. Khmelevsky. Code generation and parallel code execution from business uml models: A case study for an algorithmic trading system. In *Science and Information Conference (SAI), 2015*, pages 84–93, July 2015.

[17] F., Shadrach, B., Lacey, A.S., Roberts, A., Akbari, A., Thompson, S., Ford, D.V., Lyons, R.A., Rees, M.I. and Pickrell, W.O., “Using natural language processing to extract structured epilepsy data from unstructured clinic letters.” International Journal of Population Data Science 3.4 (2019).

[18] Weston, J.E., Bordes, A., Lebrun, A. and Raison, M.J., Facebook Inc, 2017 “Techniques to predictively respond to user requests using natural language processing.” U.S. Patent Application No. 15/077,814.

[19] Nie P, Li JJ, Khurshid S, Mooney R, Gligoric M. “Natural language processing and program analysis for supporting todo comments as software evolves.” Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence. 2018.

[20] Collobert R, Weston J, Bottou L, Karlen M, Kavukcuoglu K, Kuksa P. “Natural language processing (almost) from scratch.” Journal of machine learning research. 2011;12(Aug):2493-537.

[21] Dreisbach, C., Koleck, T. A., Bourne, P. E., & Bakken, S. “A systematic review of natural language processing and text mining of symptoms from electronic patient-authored text data.” International Journal of Medical Informatics (2019).

[22] E. Altay and M. H. Satman, "Stock market forecasting: artificial neural network and linear regression comparison in an emerging market," *Journal of Financial Management & Analysis,* vol. 18, no. 2, p. 18, 2005.

[23] L.-J. Cao and F. E. H. Tay, "Support vector machine with adaptive parameters in financial time series forecasting," *IEEE Transactions on neural networks,* vol. 14, no. 6, pp. 1506-1518, 2003.

[24] K.-j. Kim, "Financial time series forecasting using support vector machines," *Neurocomputing,* vol. 55, no. 1-2, pp. 307-319, 2003.

[25] M. Kumar and M. Thenmozhi, "Forecasting stock index movement: A comparison of support vector machines and random forest," 2006.

[26] D. M. Nelson, A. C. Pereira, and R. A. de Oliveira, "Stock market's price movement prediction with LSTM neural networks," in *Neural Networks (IJCNN), 2017 International Joint Conference on*, 2017, pp. 1419-1426: IEEE.

[27] F. Li and H. Jagadish. Constructing an interactive natural language interface for relational databases. Proceedings of the VLDB Endowment, 8(1):73{84, 2014.

[28] D. Chen and C. D. Manning. A fast and accurate dependency parser using neural networks. In EMNLP, pages 740{750, 2014.

[29] Z.Wu and M. Palmer. Verbs semantics and lexical selection. In Proceedings of the 32nd annual meeting on Association for Computational Linguistics, pages 133{138. Association for Computational Linguistics, 1994.

[30] P. Liang. Learning executable semantic parsers for natural language understanding. Commun. ACM, 59(9):68–76, Aug. 2016.

[31] L. R. Tang and R. J. Mooney. Using multiple clause constructors in inductive logic programming for semantic parsing. In Proceedings of the 12th European Conference on Machine Learning, EMCL ’01, pages 466–477, London, UK, UK, 2001. Springer-Verlag.