Natural Language to SQL Conversion

Muhammad Haseeb Ahmad*, Shivraj Mangaroliya †, Mohit Gadhiya‡
*Ahmad99@uwindsor.ca
†mangaros@uwindsor.ca
‡gadhiyam@uwindsor.ca

***Abstract*—This project focuses on enhancing task automation through the use of natural language processing (NLP). Specifically, it introduces a comprehensive system to convert natural language into structured query language (SQL). The goal is to improve the efficiency of NLP in various domains by addressing the complexities of transforming language into structured queries. The system employs methodical steps, such as standardizing expressions through root form conversion, refining vocabulary by removing stop words, and ensuring syntactical precision through SQL query structure definition. The project aims to improve clarity and precision in user query interpretation, contributing to the broader integration of NLP techniques for automating complex data retrieval tasks.**

I. Introduction and Motivation

In the initial phase of our project, the conversion of natural language to SQL is facilitated through a systematic three-step process. Commencing with the establishment of a connection to the user's specified database, the system lays the groundwork for subsequent interactions. Once this connection is successfully established, users are prompted to articulate their queries in plain language. The ensuing preprocessing steps play a pivotal role, starting with the conversion of user queries to lowercase, mitigating potential case biases. Following this, the system employs word tokenization to identify and eliminate irrelevant terms, enhancing the precision of subsequent processing. The application of Part-of-Speech (POS) tagging further refines the understanding by categorizing nouns, verbs, and adjectives, ultimately honing in on the user's query focus. The culmination of these steps involves the creation of a dictionary to store the corpus of synonyms for query words, enriching the system's linguistic comprehension. To facilitate the extraction of relevant information from the database, the project employs strategic measures. Column names are stored in a dictionary, enabling the identification of user-desired features within the database. This approach enhances the system's efficiency by providing a structured reference for interpreting user queries. Moreover, the inclusion of synonyms for column names in the dictionary introduces a dynamic element to the query process. Users can input varied terms related to the dictionary, allowing the system to intelligently identify column names through the key-value attributes of the dictionary. The motivation driving the "Natural Language to SQL" project lies in the fundamental aspiration to augment the efficiency and accessibility of database query processing. By developing a sophisticated system capable of translating user-friendly natural language queries into precise SQL commands, the project seeks to democratize access to complex databases. The integration of advanced techniques, including case normalization, POS tagging, and synonym dictionaries, underscores a commitment to overcoming the inherent challenges in linguistic interpretation. Ultimately, the project aims to empower users by simplifying interactions with databases, making data extraction more intuitive, and bridging the gap between linguistic expression and structured query language.

II. Background Study and Related Works

In today's digital age, the ubiquitous use of personal devices and internet connectivity has led to a surge in users seeking information stored in databases. However, individuals lacking expertise in database languages face challenges in accessing this data. Addressing this need, the paper proposes a Natural Language Processing (NLP) system enabling users to input structured questions in natural language and receive SQL queries, streamlining access to railways reservation database information. Leveraging techniques like tokenization, lemmatization, and parts of speech tagging, the system achieves an impressive accuracy on a dataset of 2880 natural language queries. In the context of related work, the paper draws on earlier efforts, such as Woods' 1972 system for moon rock sample information 2 retrieval, the Lifer/Ladder system's semantic grammar in 1978, and recent studies by Akshay et al., Prasun Kanti et al., and K. Javubar et al. These contributions collectively pave the way for advancing NLP-driven systems in database interaction, with the proposed railway reservation system adding specificity and efficacy to the field.[1] The test minning paper evaluates seven open-source tokenization tools, such as Nlpdotnet and NLTK, for text mining preprocessing. Nlpdotnet Tokenizer demonstrates superior performance compared to others in terms of output quality. The analysis sheds light on tool limitations, file formats, and language support, offering valuable insights for text mining applications. [2]An other paper discusses the toolkit's recent simplifications and how it enhances teaching in NLP. The paper outlines NLTK's support for tasks like tokenization, stemming, tagging, chunking, and parsing. Emphasizing its utility for students and researchers, it highlights NLTK's effectiveness in both educational and practical applications. [3]

III. Proposed Model

Below are the preprocessing steps used for generating the query from the users input.

A. Tokenization

Tokenization, a crucial process for context extraction in Natural Language Processing, plays a pivotal role in this

project. The system initiates interaction by prompting users to input queries in straightforward natural language. Subsequently, a word tokenizer is employed to break down the sentence into individual words, transforming raw text into a structured format. This structured representation facilitates the identification of user-specified features and attributes, serving as a foundational step in understanding and processing the input for subsequent stages in the project.

B. Stop words Removal

In the process of extracting specific data from the database and generating SQL queries, the significance of tokenized words cannot be overstated. Each word holds valuable information crucial for guiding the system in retrieving data from the database. However, the English language includes stop words—words that lack individual meaning. To streamline the data retrieval process, these stop words are eliminated from the query, ensuring a more efficient and focused approach to obtaining relevant information.

C. POS Tag

Part-of-Speech (POS) tags play a pivotal role in linguistic analysis by assigning grammatical categories, including nouns, verbs, and adjectives, to words within a sentence. These tags serve as a linguistic roadmap, offering valuable insights into the structural composition of sentences. They contribute significantly to language understanding, enabling machines to parse sentences accurately and discern the roles and relationships of words within a given context. The importance of POS tags extends to various natural language processing tasks, such as machine translation, information retrieval, and sentiment analysis. By providing a nuanced understanding of word functions, POS tags empower machines to navigate the complexities of language, facilitating the development of more sophisticated language models. This, in turn, enhances the overall efficacy of computational systems engaged in linguistic analysis.

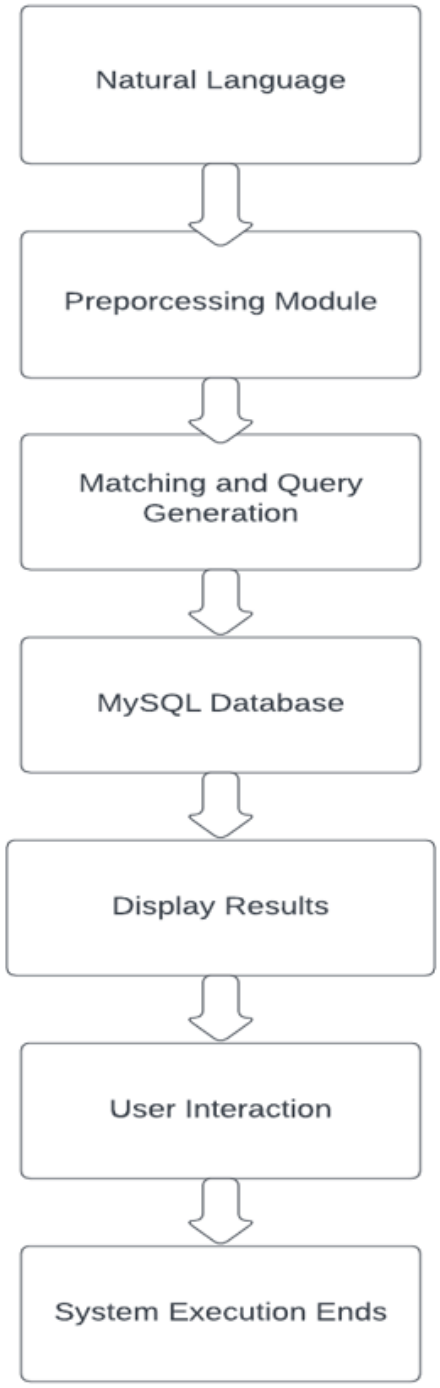
D. Lower Case

In the pursuit of mitigating biases inherent in textual data, a crucial preprocessing step involves converting user queries to lowercase. This deliberate choice is designed to foster uniformity and diminish potential ambiguities, thereby amplifying the clarity of the requested data. By enforcing this standardized approach, the system ensures a more dependable and impartial interpretation of user input, ultimately elevating the effectiveness of the data analysis and retrieval process to achieve more accurate and equitable results.

E. Dictionary

In the context of crafting SQL queries, certain ubiquitous words like ”Select,” ”From,” and ”Update” play pivotal roles. To streamline the SQL query generation process, we’ve meticulously curated a comprehensive dictionary housing synonyms for these essential SQL terms. This strategic compilation empowers the system to adeptly interpret user input, facilitating a nuanced understanding and seamless generation of SQL queries based on the synonyms encapsulated within the established dictionary.

MODEL



The block diagram serves as an illuminating depiction of the intricate architecture and operational sequence of a natural language processing system meticulously tailored for

querying a MySQL database. Commencing with user input in natural language, the system embarks on a preprocessing module to refine and structure the query. This module employs tokenization and stopwords removal, contributing to a more streamlined and meaningful representation of the user's input. Following this preparatory phase, the refined query undergoes a critical matching process with tables within the MySQL database. The outcome of this matching endeavor is the generation of SQL query suggestions, effectively translating the user's natural language query into a structured format understood by the database. Post-matching and query generation, the system seamlessly transitions to the database connection phase. Here, the MySQL database, serving as the repository of structured information, facilitates the retrieval of relevant data based on the generated SQL queries. The interconnected nature of the system enables a dynamic feedback loop, presenting the matched tables, suggested SQL queries, and any pertinent information to the user. This user-friendly interface not only expedites the query process but also provides a platform for iterative refinement. Users can adapt their queries based on feedback, fostering a more intuitive and interactive experience. Crucially, the block diagram encapsulates the system's iterative functionality, emphasizing the ongoing interaction between the user and the system. This iterative approach facilitates user engagement by allowing them to receive feedback on the matched tables and adjust their queries accordingly. The visual representation of these processes enhances the comprehensibility of the system's core functionalities, demonstrating its adaptability and responsiveness to user input throughout the entire query process.

IV. Results and Figures

As per the specified model, system ask for an input in natural English language. Based on the input it process the input. By following complete procedure of generating query, it generates the SQL query and execute it on MySQL. Some of test cases are provided as below:

Example 1: User Input : Give me firstname and lastname of employee whose departmenttype is Sale

system resulting query : SELECT firstname,lastname from employee_data WHERE departmenttype = 'Sales'

```
Enter your question:give me firstname and lastname of employee whose departmenttype is Sales
['give me firstname and lastname of employee ', ' departmenttype is Sales']
WHERE departmenttype = 'Sales'
SELECT firstname, lastname FROM employee_data
SELECT firstname, lastname FROM employee_data WHERE departmenttype = 'Sales'
Results for simple query:
('Edward', 'Buck')
('Michael', 'Riordan')
('Jasmine', 'Onque')
('Maruk', 'Fraval')
('Latia', 'Costa')
('Sharlene', 'Terry')
('Jac', 'McKinzie')
('Joseph', 'Martins')
('Myriam', 'Givens')
('Dheepa', 'Nguyen')
('Bartholemew', 'Khemmich')
('Xana', 'Potts')
('Prater', 'Jeremy')
```

Example 2: User Input : Give me firstname and lastname and dob of employee

system resulting query : SELECT firstname, lastname dob from employee_data

```
Enter your question:give me firstname and lastname and dob of employee
['give me firstname and lastname and dob of employee']
SELECT firstname, lastname, dob FROM employee_data
SELECT firstname, lastname, dob FROM employee_data
Results for simple query:
('Uriah', 'Bridges', '07-10-1969')
('Paula', 'Small', '30-08-1965')
('Edward', 'Buck', '06-10-1991')
('Michael', 'Riordan', '04-04-1998')
('Jasmine', 'Onque', '29-08-1969')
('Maruk', 'Fraval', '03-04-1949')
('Latia', 'Costa', '01-07-1942')
('Sharlene', 'Terry', '07-03-1957')
('Jac', 'McKinzie', '15-05-1974')
('Joseph', 'Martins', '11-11-1949')
('Myriam', 'Givens', '26-01-1964')
('Dheepa', 'Nguyen', '06-04-1948')
('Bartholemew', 'Khemmich', '24-11-1981')
('Xana', 'Potts', '06-11-1951')
```

Example 3: User Input : Give me firstname of employee whose dob is 01-07-1942

system resulting query : SELECT firstname from employee_data WHERE dob = '01-7-1942'

```
Enter your question:give me firstname of employee whose dob is 01-07-1942
['give me firstname of employee ', ' dob is 01-07-1942']
WHERE dob = '01-07-1942'
SELECT firstname FROM employee_data
SELECT firstname FROM employee_data WHERE dob = '01-07-1942'
Results for simple query:
('Latia',)
('Janiyah',)
```

Example 4: User Input : show me all data

system resulting query : SELECT * from employee_data

```
Enter your question:show me all data
['s', ' me all data']
Invalid conditional query format.
SELECT * FROM employee_data
SELECT * FROM employee_data
Results for simple query:
(3427, 'Uriah', 'Bridges', '20-Sep-19', '', 'Production Technician I', 'Peter Oneill', 'uriah.bridges@bilearner.com', 'CCDR',
'Active', 'Contract', 'Zone C', 'Temporary', 'Unk', '', 'Production       ', 'Finance & Accounting', '07-10-1969', 'MA', 'Acc
ounting', 'Female', 34904, 'White', 'Widowed', 'Fully Meets', 4)
(3428, 'Paula', 'Small', '11-Feb-23', '', 'Production Technician I', 'Renee Mcconmick', 'paula.small@bilearner.com', 'EW', '
ctive', 'Contract', 'Zone A', 'Part-Time', 'Unk', '', 'Production       ', 'Aerial', '30-08-1965', 'MA', 'Labor', 'Male', 65
3, 'Hispanic', 'Widowed', 'Fully Meets', 3)
(3429, 'Edward', 'Buck', '10-Dec-18', '', 'Area Sales Manager', 'Crystal Walker', 'edward.buck@bilearner.com', 'PL', 'Activ
e', 'Full-Time', 'Zone B', 'Part-Time', 'Unk', '', 'Sales', 'General - Sga', '06-10-1991', 'MA', 'Assistant', 'Male', 2330,
'Hispanic', 'Widowed', 'Fully Meets', 4)
(3430, 'Michael', 'Riordan', '21-Jun-21', '', 'Area Sales Manager', 'Rebekah Wright', 'michael.riordan@bilearner.com', 'CCD
R', 'Active', 'Contract', 'Zone A', 'Full-Time', 'Unk', '', 'Sales', 'Finance & Accounting', '04-04-1998', 'ND', 'Clerk', 'M
le', 58782, 'Other', 'Single', 'Fully Meets', 2)
(3431, 'Jasmine', 'Onque', '29-Jun-19', '', 'Area Sales Manager', 'Jason Kim', 'jasmine.onque@bilearner.com', 'TNS', 'Activ
```

V. Limitation and Challenges

The project grapples with the formidable challenge of designing a rule-based system, encountering complexities in handling synonymous expressions within the English language, which necessitates a robust approach for generalization tailored to SQL generation. Synonyms pose a notable challenge, demanding careful consideration to ensure accurate and versatile rule application. Furthermore, the extraction of information from SQL databases introduces intricacies, particularly in crafting complex queries involving the left join, right join, Outer join, Union and similar constructs adds an additional layer of complexity to the system's development.

Notably, the system's design imposes a specific structuring constraint when composing text in natural language. Users must adhere to a prescribed sequence to get expected output. This unique approach ensures clarity in communication with the system and underscores the intricacies inherent in facilitating seamless interaction between natural language input and SQL query generation.

Sometimes user can input small sentences. Those English statements are less informative to extract SQL query, as an example who is Alice? So basically system can not generate the SQL query for this sentence and can not provide information about Alice.

VI. Future work

The primary objective of this project is to facilitate user interaction with a SQL database by employing natural language queries for data extraction. As the system evolves, we've identified key challenges that present opportunities for refinement in subsequent iterations. A crucial area for improvement lies in the extension of the rule-based system governing data extraction. Currently, the system demonstrates efficacy when attribute information aligns with the specified feature sequence. However, deviations from this prescribed order result in the system generating messages indicating the inability to locate the requested record. Addressing this limitation is paramount to enhancing user experience and ensuring the system's adaptability to diverse query structures. Moreover, the project stands to gain substantial improvements through the expansion of its lexical database to incorporate synonyms. By broadening the system's linguistic repertoire, it can better interpret and respond to user queries, accommodating variations in expression while maintaining accuracy. The inclusion of synonyms is pivotal in elevating the system's capacity to deliver comprehensive and precise results, aligning more closely with user expectations. This expansion aligns with the project's overarching goal of fostering a seamless and intuitive interaction between users and the SQL database, promoting efficiency and user satisfaction in information retrieval processes.

VII. Conclusion

In conclusion, the "Natural Language to SQL" project represents a significant stride towards democratizing access to databases by transforming user-friendly natural language queries into precise SQL commands. The systematic three-step process, encompassing connection establishment, user query preprocessing, and database information extraction, is fortified by advanced techniques such as case normalization, POS tagging, and synonym dictionaries. The model's robust architecture, illustrated through the block diagram, ensures an iterative and interactive user experience, allowing for feedback and query refinement. Results indicate the system's proficiency in generating accurate SQL queries from diverse natural language inputs. However, challenges, including handling synonymous expressions and structuring constraints, underscore the ongoing complexities in linguistic interpretation. Despite these challenges, the project showcases a commitment to advancing NLP-driven systems, marking a significant contribution to the evolving landscape of database interaction and paving the way for more intuitive and accessible data retrieval processes.

References

1 Uma, M., Sneha, V., Sneha, G., Bhuvana, J., and Bharathi, B., "Formation of sql from natural language query using nlp," 02 2019, pp. 1–5.

2 Mohan, V., "Text mining: Open source tokenization tools: An analysis," vol. 3, pp. 37–47, 01 2016.

3 Bird, S., "Nltk: The natural language toolkit," 01 2006.