IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 7, April 2024

Natural Language to SQL Queries Generation using NLP Techniques

Mohammed Osama Shaikh[1], Fiza Shaikh[1], Dr. Irfan Landge[2]
Students, Department of Information Technology Engineering[1]
Assistant Professor, Department of Information Technology Engineering[2]
M.H. Saboo Siddik College Of Engineering, Mumbai, Maharashtra, India

Abstract: *Databases are progressively standard components of modern websites and applications. They are regularly utilized by those who need an in-depth understanding of their structure and impressive competence in the field. Hence, these days it has ended up exceptionally imperative to build a framework that can interpret common dialect to SQL questions. Be that as it may, because of restrictions in their design, numerous existing frameworks are restricted to specific databases. In this paper, we have shown a modern approach that employs English dialect inquiries to any database. The result demonstrates the effectiveness of this approach by retrieving the information from the database and giving the output to the user query.*

Keywords: Database (DB), SQL queries, English SQL translator, Human-DB interface

I. INTRODUCTION

Natural Language Processing (NLP) has become a very important field in recent years, as it enables systems to understand as well as process human language in a more natural way. One of the most valuable applications of NLP is the ability to generate SQL queries from natural language inputs, which can greatly simplify the process of accessing and analyzing data stored in databases.

The project "Natural Language to SQL Queries Generation using NLP techniques" aims to develop a system that can automatically convert natural language input into SQL queries. This system will leverage various NLP techniques, such as tokenization, stopword removal, lemmatization, part-of-speech tagging, and parsing, to analyze and understand the user's input. By mapping the identified entities and attributes to the corresponding database schema, the system will be able to construct the appropriate SQL query to retrieve the desired information.

The potential applications of such a system are vast, as it can be integrated into various domains where the non-technical user can communicate with databases without having to learn the complexity of SQL. For instance, it could be used in business intelligence tools, data analytics platforms, or even conversational interfaces, enabling users to access and analyze data using natural language queries[1]. Additionally, this project can contribute to the ongoing research in the field of NLP and database systems, potentially leading to further advancements and improvements in this area[2].

II. TRANSLATING NATURAL LANGUAGE TO SQL QUERIES

A database is a structured collection of data stored and organized on a computer system. In a relational database, information is stored in the form of tables, which are matrix-like structures. A relational database can consist of one or more tables, which may or may not be linked together. The data entries (information) within a table are organized into columns (also called fields). A group of columns relating to the same entity (object) forms a table.

A schema, also known as a data model, is a diagrammatic or textual description that specifies the distribution and organization of data within a database. It provides information on the characteristics of each data type and the relationships between them. A relational schema is the most commonly used approach for describing a relational database.

SQL (Structured Query Language) is a standardized programming language designed for performing operations on relational databases. The objective of this project is to propose an application that allows querying a relational database

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 7, April 2024

using natural language questions in English. The focus is solely on querying the database, which is accomplished using the SELECT command in SQL, the syntax is as follows:

```
SELECT column_list
FROM table_list
[JOIN joint_expression]
[WHERE conditional_expression]
[GROUP BY group_by_column_list]
[HAVING conditional_expression]
[ORDER BY order_by_column_list]
```

When the user enters an input like:

What is **the age** of a **student** whose first **name is Karan**?

The system will execute the query like this:

SELECT **age** FROM **students** WHERE **first name** = 'KARAN'

III. EXISTING SYSTEM

Traditionally, retrieving information from databases has relied on manually crafting SQL queries, which requires a comprehensive understanding of SQL and the underlying database schema [1]. This presents a significant barrier for non-technical users, hindering their ability to access and analyze data effectively [10]. To address this challenge, techniques have been developed for automatically translating natural language queries into SQL statements by leveraging natural language processing (NLP) methods [3,7]. A prominent approach uses Probabilistic Context-Free Grammars (PCFGs) to model the syntactic structure of natural language and map it to corresponding SQL constructs [4,8].

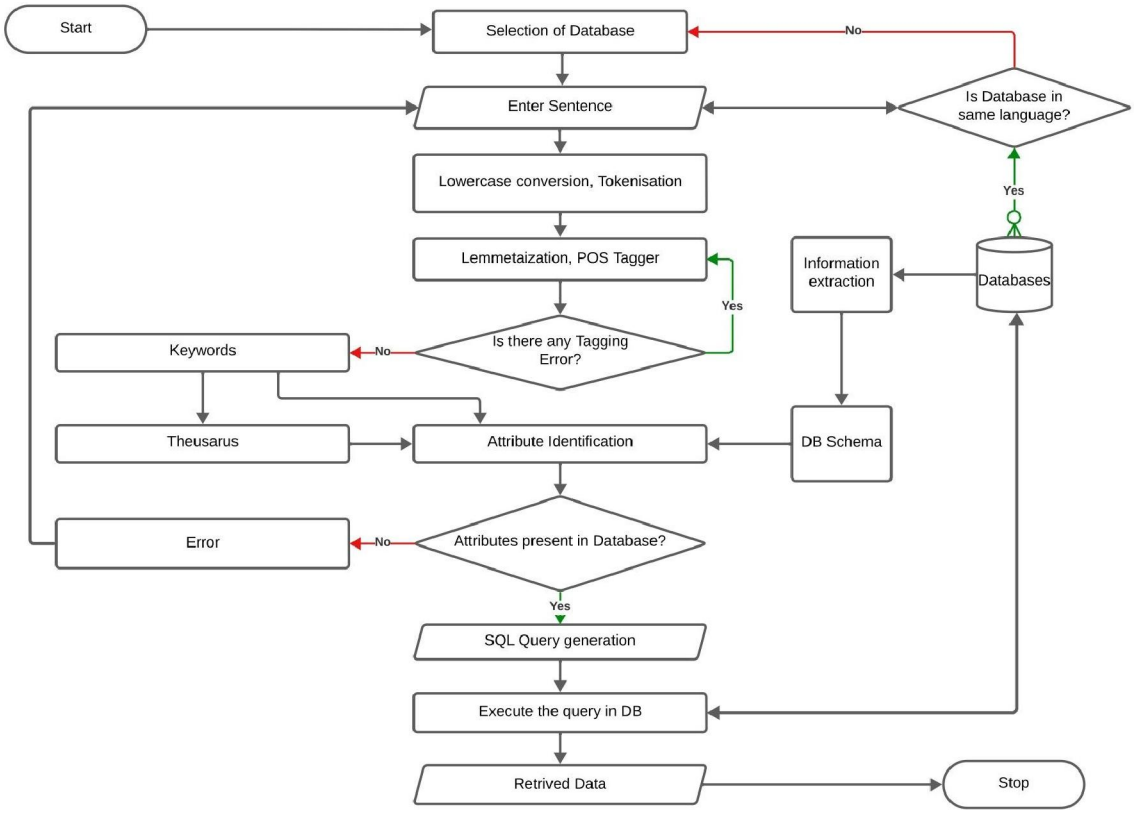


Fig: Diagram representing the synthesized architecture of the operation of the NL2SQL system

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 7, April 2024

Open-source frameworks like Stanford CoreNLP have adopted PCFG-based approaches for translating natural language to SQL [4]. However, such systems face limitations like the need to manually curate stop word lists
[5] and challenges in handling complex linguistic constructs across multiple languages [6]. While they show promise for simple queries, their performance degrades for more complex cases involving nested clauses, multiple conditions, and advanced SQL features [2]. This highlights the need for more robust NLP techniques to accurately capture nuances while maintaining an understanding of database schemas.

To overcome these challenges, researchers are exploring alternative deep learning approaches like transformer architectures and pre-trained language models [9], which have demonstrated remarkable success across various NLP tasks. Integrating these advanced models with techniques like transfer learning and domain adaptation holds the potential for improving the accuracy and robustness of natural language to SQL translation systems [3]. This would enable seamless handling of increasingly complex queries across diverse domains without compromising user accessibility.

IV. PURPOSE SYSTEM ARCHITECTURE

The system architecture consists of eight key steps:

Lowercase Conversion:

Convert the user's input to lowercase This step ensures consistency in the input by converting all characters to lowercase. Maintaining case consistency is crucial for NLP tasks as it helps in accurately identifying and processing words and entities. By converting the input to lowercase, the system can handle variations in capitalization and treat words like "Apple" and "apple" as the same[3].

Tokenize the user input:

The input is divided into discrete words or phrases (tokens) at this step. As it divides the input text into meaningful units (tokens) for additional processing, tokenization is a crucial stage in natural language processing (NLP). Depending on the demands of the work, several tokenization approaches, such as word tokenization, sentence tokenization, or subword tokenization, might be used[4].

Example:

Input: "What are the names of students enrolled in the Information Technology program?"

Tokenization: ["What", "are", "the", "names", "of", "students", "enrolled", "in", "the", "Information", "Technology", "program?"]

Stop words removal:

In this step, common words that don't have any meaning—like "the," "is," and "and"—are eliminated. Words that are commonly used in everyday speech but have little bearing on the text's overall meaning are known as stopwords. Eliminating these terms can assist lower the noise and increase the precision of the processes that come after[5].

Example:

Tokenized input: ["What", "are", "the", "names", "of", "students", "enrolled", "in", "the", "Information", "Technology", "program?"]

After stopword removal: ["What", "names", "students", "enrolled", "Information", "Technology", "program?"]

Lemmatize the tokens:

Words are reduced to their simplest form in this step (for example, "running" becomes "run"). Lemmatization is the process of organizing a word's inflected forms into a single group so that they can be examined as a unit. This can enhance the accuracy of the following processes and help comprehend the input's semantic meaning more fully[6].

Example:

Input tokens: ["names", "students", "enrolled", "Information", "Technology", "program?"]

After lemmatization: ["name", "student", "enroll", "information", "technology", "program"]

Assign parts of speech:

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 7, April 2024

In this step, the tokens are given grammatical roles (such as verbs, nouns, etc.). The first stage in comprehending the input text's grammatical structure is part-of-speech (POS) tagging. The system can better understand the relationships between words and their roles in sentences by applying POS tags to the tokens[7].

Example:

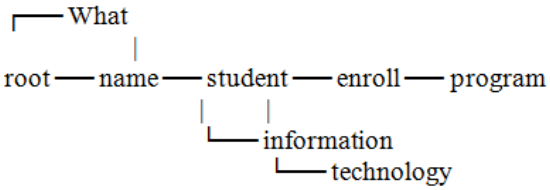
Tokens: ["name","student", "enroll", "information", "technology", "program"]

POS tags: ["NN", "NNS", "VB", "NN", "NN", "NN"]

Analyze the grammatical structure:

This step analyzes the relationships between words and phrases in the user's input. Detailed: Understanding the grammatical structure of the input is essential for accurately interpreting the user's intent and mapping the input to the corresponding database schema. This step involves techniques like dependency parsing or constituency parsing to analyze the syntactic structure of the input[8].

Example:



Identify the key attributes and entity:

This step identifies the relevant attributes and entities mentioned in the user's input that should be mapped to the database schema. Detailed: Identifying the key attributes and entities is a crucial step in translating the natural language input into an SQL query. This step involves techniques like named entity recognition (NER) and relation extraction to identify and classify the relevant entities and their relationships[9].

Example:

Input: "What are the names of students enrolled in the Information Technology program?"

Identified Entities:

- Student
- Program: Information Technology

Map the attributes and entities to the database schema:

In this stage, the detected entities and characteristics are mapped to the relevant database schema tables and columns. Following their identification, the important properties and entities must be mapped to the relevant database schema tables and columns. This stage necessitates a deep comprehension of the database structure as well as the capacity to create precise mappings between the parts of the schema and the natural language input[10].

Example:

Student Table:

- student_id (PK)
- first_name
- last_name

Program Table:

- program_id (PK)
- program_name

Enrollment Table:

- student_id (FK)
- program_id (FK)

Finally, the system constructs an SQL query based on the mapping and the user's input and displays the generated query on the screen.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 7, April 2024

V. DRAWBACKS

While the proposed system offers several advantages, some potential drawbacks should be considered:

- **Natural Language's Complexity:** It can be difficult to faithfully convert natural language into a SQL query since natural language is inherently ambiguous and can have numerous meanings of the same query. It can be especially challenging to handle intricate verbal constructions like conditionals, negations, and conjunctions.
- **Reliance on domain expertise:** The availability of domain expertise and the capacity to transfer natural language items to the proper database schema elements may have a significant impact on the system's performance. This may be somewhat difficult, particularly in fields where database structures are intricate or dynamic.
- **Managing Contextual Data:** It can be challenging to include contextual information from natural language queries—like answers to earlier questions or common knowledge—into the translation process**.** Neglecting to consider the context may result in inaccurate translations or misconstrued meanings.
- **Managing Non-Vocabulary Words:** Translation mistakes may result from the system's inability to handle terms or entities that are not included in its training set or knowledge base.
- **Efficiency and Scalability:** The computing requirements for the translation process may become prohibitive as the complexity of the database schema or the natural language input rises, restricting the scalability and efficiency of the system.

VI. CONCLUSION

The proposed system leverages various Natural Language Processing (NLP) techniques, such as tokenization, stopword removal, lemmatization, part-of-speech tagging, and parsing, to analyze and understand the user's natural language input. It then identifies the key entities and attributes mentioned in the query and maps them to the corresponding database schema elements. Finally, it constructs an SQL query based on this mapping and displays it to the user.

This approach has the potential to significantly improve the accessibility of databases for non-technical users, enabling them to query and analyze data using natural language rather than having to learn and write complex SQL queries. However, the translation process faces several challenges, such as handling the ambiguity and complexity of natural language, accounting for contextual information, and dealing with out-of-vocabulary words or entities.

Notwithstanding these obstacles, continuous research endeavours in the domains of natural language processing and database systems seek to tackle these constraints and enhance the precision, resilience, and expandability of these systems. Natural language queries may be increasingly often and reliably translated into SQL queries as NLP techniques evolve and more effective algorithms and models are created.

REFERENCES

[1]. Zafar, A., Malik, M. S. I., & Suriadi, S. (2020). “Natural language to SQL query translation: A novel approach for user interpretation. Information Systems”.

[2]. Giannotti, F., Nanni, M., Pedreschi, D., Pinelli, F., Renso, C., Rinzivillo, S., & Trasarti, R. (2011). “Unveiling the complexity of human mobility by querying and mining massive trajectory data”.

[3]. Bird, S., Klein, E., & Loper, E. (2009). Natural Language Processing with Python. O'Reilly Media, Inc

[4]. Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., & McClosky, D. (2014). “The Stanford CoreNLP natural language processing toolkit”. In Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations.

[5]. Wilbur, W. J., & Sirotkin, K. (1992). “The automatic identification of stop words. Journal of information science”.

[6]. Plank, B., Søgaard, A., & Goldberg, Y. (2016). “Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss”.

[7]. Jurafsky, D., & Martin, J. H. (2009). “Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition”.

[8]. Klein, D., & Manning, C. D. (2003). “Accurate unlexicalized parsing. In Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics”.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 7, April 2024

[9]. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., & Dyer, C. (2016). "Neural architectures for named entity recognition".

[10]. Li, F., & Jagadish, H. V. (2014). "Constructing an interactive natural language interface for relational databases".