CIS*6050 Final Project Report

Natural Language to SQL Converter

Nidish Murugan
Department of Mathematics
and Statistics
University of Guelph
Guelph Canada
nmurugan@uoguelph.ca

Suraj Karthik S A
Department of Mathematics
and Statistics
University of Guelph
Guelph Canada
ssunanda@uoguelph.ca

Sonal Sushil Gupta
Department of Mathematics
and Statistics
University of Guelph
Guelph Canada
sonalsus@uoguelph.ca

ABSTRACT

This project presents a comprehensive analysis of the performance of implemented models, including the T5 transformers and the LLM model, in generating SQL queries based on the WikiSQL dataset. Beginning with a rigorous evaluation utilizing devised metrics, valuable insights into the effectiveness of these models were obtained. Promising capabilities in accurately generating SQL queries were observed, with the LLM model demonstrating notable performance enhancements through outlined manipulation steps. Standardizing the output and aligning it with WikiSQL labeling conventions ensured fair and robust comparisons against ground truth. These results underscore the potential of advanced natural language processing techniques for precise query generation. The T5 transformers achieved an accuracy of 58.06%, while the LLM model achieved a higher accuracy of 64.19%. These findings highlight varying degrees of efficacy in capturing the semantics and nuances of input questions. Additionally, the LLM model outperformed the T5 transformers in specific query operations, particularly aggregation, union, and join operations, showcasing its capability to excel in complex query scenarios and enhance the efficiency and accuracy of query generation systems in real-world applications.

Keywords

Natural Language, SQL Query, Neural Networks, Transformers, Intent Detection, Entity Recognition

INTRODUCTION

Companies have large amounts of data in their database and employees in the company should be able to fetch and query data as per company policies. However, to fetch this data employees often are dependent on teams who perform this basic task to write SQL queries and give them the data as all employees do not have the technical knowledge. To bridge this gap, we propose the idea of converting natural language text to SQL which will not only ease the process of querying data but also enhance the accessibility of the databases to users across multiple domains who have limited extensive technical knowledge.

To mitigate these challenges and democratize database accessibility, we propose a transformative solution: a Natural Language to SQL (NL2SQL) conversion system leveraging advanced neural network architectures. The primary objective of this project is to empower users across diverse domains, irrespective of their technical proficiency, to interact with databases seamlessly using natural language queries. By bridging the gap between technical and non-technical stakeholders, this system eliminates the need for intermediary translation, thereby enhancing overall database accessibility and usability.

The motivation behind this research stems from the recognition of the pressing need for intuitive and user-friendly interfaces for database querying. Traditional SQL interfaces, while powerful, present formidable barriers to entry for non-technical users, hindering effective collaboration and knowledge sharing within organizations. By enabling users to articulate their data requirements in natural language, our NL2SQL system aims to democratize database access, fostering a more inclusive and collaborative data-driven culture within enterprises.

This project seeks to address the following questions:
1. How can advanced neural network architectures, such as transformers and Large Language Models (LLMs), be effectively leveraged to convert natural language queries into SQL statements?
2. What methodologies and techniques can optimize the accuracy and efficiency of NL2SQL conversion, considering the diverse range of database schemas and query types?

The importance of this problem is that it could completely change the landscape of database querying, democratizing access to critical organizational data and empowering users with varying levels of technical expertise. Beyond organizational contexts, NL2SQL conversion has broad applications in domains like business intelligence, data analytics, where intuitive and efficient data querying is paramount for informed decision-making and strategic planning.

PROBLEM DEFINITION

The problem at hand is to develop a Natural Language to SQL (NL2SQL) conversion system that accurately translates a given natural language query Q into a corresponding SQL query S representing the user's intent in querying a database. This task requires designing a model that is capable of mapping natural language queries to SQL queries with high accuracy and efficiency, thereby enabling users to interact with databases using familiar language constructs.

There are several constraints and restrictions that need to be considered in this problem domain. Firstly, the generated SQL query S must comply with the syntax rules of the SQL language to ensure its executability on the database system. Additionally, the SQL query S should accurately capture the user's intent as expressed in the natural language query Q, ensuring semantic accuracy and relevance in the generated output.

The optimization objective in this problem is to maximize the accuracy of the NL2SQL conversion system. Accuracy is defined in terms of the precision and recall of the generated SQL queries compared to ground truth SQL queries, ensuring that the model produces highly accurate translations that faithfully represent the user's information needs.

However, the NL2SQL conversion problem is inherently challenging due to various factors. Firstly, natural language understanding introduces complexities stemming from syntactic and semantic ambiguities inherent in human language. Resolving these ambiguities and accurately mapping natural language expressions to structured SQL queries require sophisticated modeling approaches. Additionally, the diversity of database schemas and query types further complicates the problem, necessitating robust and flexible modeling techniques to handle a wide range of query scenarios effectively.

BACKGROUND AND RELATED WORK

SQL query generation, the process of automatically generating structured query language (SQL) statements from natural language questions or queries, is a crucial component in database management systems and natural language processing (NLP) applications. The task involves translating human-readable questions into executable SQL queries, facilitating seamless interaction between users and databases. Traditional approaches to SQL query generation often rely on rule-based systems or handcrafted templates, which may lack flexibility and struggle to handle complex queries or variations in natural language input.

He et al.'s [4] study highlights the dominance of LSTM-based models in natural language to SQL query conversion, exploring the potential of transformers and CNN architectures for enhanced accuracy. The study integrates BERT and Glo-Ve embeddings to emphasize syntactical structures in text-to-SQL translation. Evaluation metrics, including query-match and execution accuracy, reveal the superiority of BERT embeddings, signaling the need for structural changes. The study recognizes challenges in training the transformer encoder for SELECT_COL prediction. In summary, the study underscores the importance of model selection and linguistic understanding for improving text-to-SQL translation.

Recent advancements in deep learning and NLP have sparked interest in developing data-driven approaches to SQL query generation, offering the potential to overcome the limitations of rule-based methods. Researchers have explored various neural network architectures, including sequence-to-sequence models, transformer-based models, and language models pre-trained on large text corpora, for this task.

Kim et al.'s paper [5] introduces crucial NLP concepts, addressing RNN limitations and proposing alternatives like LSTMs, GRUs, and attention techniques. Investigating Seq2Seq models and the role of pointer networks, the study uncovers NL2SQL method limitations, emphasizing the need for advancements. Evaluation metrics expose shortcomings in current models like SyntaxSQLNet and GNN, indicating the necessity for further development in sequence modeling.

The MAC-SQL framework [6] takes a new approach to Text-to-SQL parsing by leveraging natural language processing, it employs Large Language Models (LLMs) such as GPT-4 and SQL-Llama. It proposes a multi-agent collaboration strategy, which differs from the largely single-agent models used in prior studies. By combining a core Decomposer agent with auxiliary agents for database simplification and query refinement, MAC-SQL effectively solves SQL query complexity while achieving cutting-edge execution accuracy on datasets such as BIRD. This methodology exhibits the success of previous advances in LLMs while proposing an entirely new model for multi-agent collaboration. The model [MAC-SQL+GPT-4] achieves an execution accuracy of 59.39 and Exact match accuracy of 63.20 on the dev set of BIRD and Spider with few-shot evaluation. This paper provides a solid reference on the application of LLMs on Text to SQL generation.

Sutskever et al. (2014) [7] introduced sequence-to-sequence learning with recurrent neural networks (RNNs) for machine translation, inspiring subsequent work on applying this architecture to SQL query generation. Dong Lapata (2016) [8] proposed a syntax-based approach using tree-to-sequence models to generate SQL queries from natural language questions.

Models built on transformer architecture, like the T5 model pioneered by Raffel et al. (2019), have exhibited impressive proficiency across a spectrum of natural language processing (NLP) assignments, encompassing tasks such as text creation and linguistic translation. By harnessing self-attention mechanisms, these models effectively grasp extensive dependencies and contextual nuances, rendering them particularly apt for tasks involving the generation of SQL queries.

Moreover, recent advancements in large language models (LLMs), like GPT (Generative Pre-trained Transformer) series from OpenAI's , have shown promise in generating coherent and contextually relevant text. While primarily designed for language understanding and generation, LLMs can be fine-tuned for specific tasks, including SQL query generation.

Despite these advancements, challenges remain in accurately capturing the semantics and nuances of natural language questions and translating them into precise SQL queries. Additionally, evaluating the performance of generated queries against ground truth labels presents inherent difficulties, requiring careful consideration of metrics and evaluation methodologies.

METHODOLOGY

1. Natural Language (NL) Input

Users express their queries in natural language, specifying the information they wish to derive from the dataset. These user inputs serve as the input to the model.

2. Intent Detection

In models like Transformers and Generative AI, intent detection involves categorizing queries into different types, such as SELECT, INSERT, UPDATE, DELETE, and more. This procedure depends on pretrained models, which have undergone training using datasets such as WikiSQL. These models possess the ability to comprehend the user's intent embedded within the query and subsequently categorize it accordingly.

For example, if a user asks a question about retrieving information from the database, the intent detection module will classify it as a SELECT query.

Similarly, if the user wants to update or delete data, the module will recognize the corresponding intent. This approach enables the system to accurately interpret user queries and generate appropriate SQL commands in response.

3. Entity Recognition

From the input, relevant entities such as table names, column names, and conditions are identified to construct a query. To achieve this, models were trained on annotated data containing labeled entities from the chinook dataset, enhancing recognition accuracy.

A rule-based token mapping approach is employed, which involves searching for keywords and relevant details in the input through exact matching, partial matching, conjunction/disjunction identification, and numerical aggregator identification.

Our exploration encompassed datasets like WikiSQL, Spider, and ATIS, with WikiSQL emerging as the preferred choice for testing the model due to its comprehensive coverage and meticulously defined ground truth for various query scenarios. This feature, coupled with the dataset's diversity, solidified our decision to utilize WikiSQL as the dataset for testing and evaluation purposes.

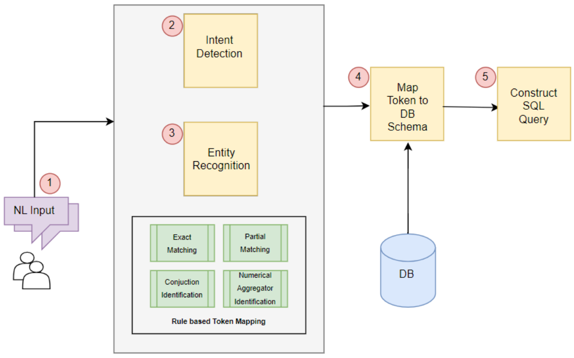


Figure 1: Architecture of the model

4. Map Tokens to DB Schema

In the data processing pipeline, after intent detection and entity recognition, the next crucial step involves mapping recognized tokens to corresponding elements in the database schema. This process ensures accurate and contextually relevant SQL query construction. A mapping mechanism associates identified entities like table and column names with their counterparts in the database structure, facilitating syntactically and semantically correct query generation. Additionally, a rule-based token mapping approach is adopted, employing techniques such as exact and partial matching. This meticulous alignment enhances query accuracy and relevance, enabling effective user-database interaction. Leveraging advanced techniques like semantic parsing enhances system robustness, particularly in handling diverse query scenarios. This streamlined approach improves overall usability, empowering users to query data effectively and intuitively.

5. Model Selection and Architecture Exploration

In our approach, we meticulously selected and explored various language models to determine their effectiveness. We extensively studied state-of-the-art models such as Generative Pre-trained Transformers (GPT), Bidirectional Encoder Representations from Transformers (BERT), and Text-To-Text Transfer Transformer (T5). Each of these models offers unique advantages in understanding context, tokenization mechanisms, and generating text.

During our experimentation phase, we specifically focused on the Transformer T5 model developed by Hugging Face. T5, or Text-To-Text Transfer Transformer, is renowned for its flexibility in handling a wide range of Natural Language Processing (NLP) tasks. Unlike traditional sequence-to-sequence models, T5 approaches tasks in a text-to-text manner, allowing it to tackle various tasks like translation, summarization, and question answering within a unified framework. However, despite its versatility, we observed that T5 struggled with accuracy, particularly in the task of converting natural language into SQL queries.

Our next step involved conducting experiments using the Generative Pre-trained Transformer (GPT) model as our main Large Language Model (LLM). Gemini AI Pro, a deep learning architecture based on the Transformer model, demonstrated exceptional capabilities in understanding natural language and in the generation of SQL Queries. Its ability to generate logical and contextually appropriate text made it an ideal candidate for Natural Language to SQL query conversion.

We then fine-tuned the pre-trained Gemini AI Pro model, which is based on the GPT architecture, using transfer learning techniques. By exposing the model to WikiSQL's labeled data, we aimed to adapt its parameters to the intricacies of Natural Language to SQL query conversion. Through this process, the model learned to capture the semantic relationships between natural language inputs and SQL query outputs, enabling accurate and contextually appropriate translations.

EVALUATION METRICS

The evaluation process involves feeding natural language queries into the language model and comparing the generated SQL queries against ground truth queries. Each generated SQL query is assessed for correctness and semantic coherence, considering factors such as syntactic structure, attribute selection, and condition accuracy. The overall accuracy is calculated in terms of the percentage of common tokens between expected and generated SQL Queries out of the total tokens in the expected SQL query.

Accuracy Calculation

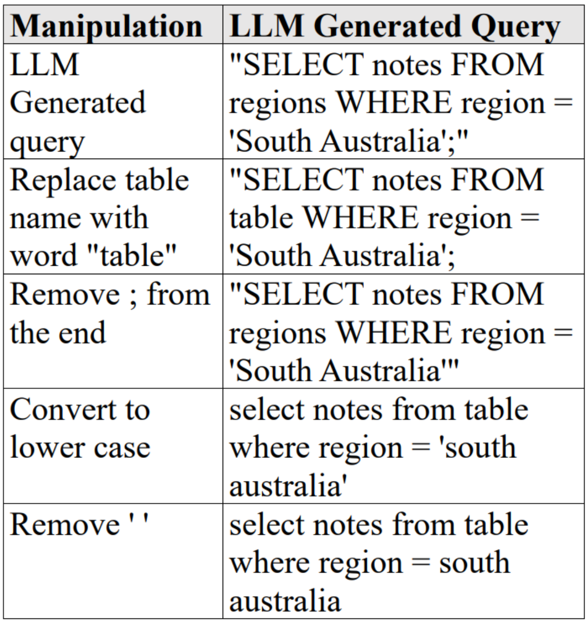
The accuracy is computed using the formula:

$$\text{Accuracy} = \frac{\text{No. of common words between GRQ and GTQ}}{\text{Total No. of words in GRQ}}$$

Where:

- GRQ (Ground Reality Query): The actual, true query fetched from the WikiSQL dataset label.
- GTQ (Generated Query): The produced query from the trained model.

To ensure a standardized evaluation process, we have implemented specific manipulations on the output of the Large Language Model (LLM) model compared to the labeled SQL queries in the WikiSQL dataset. Firstly, we prompt the model to generate the SQL query within a single line, deviating from its typical output format that spans multiple lines. This adjustment facilitates a direct and straightforward comparison between the model's output and the labeled queries, streamlining the evaluation process. Secondly, we meticulously remove any semicolons from the end of the generated query to maintain consistency with the format of the labeled queries, ensuring alignment in syntax and structure. Additionally, given that the LLM model infers table names based on the input question, we systematically replace these inferred names with the generic term "table," adhering to the standard table naming convention established within the WikiSQL dataset. Finally, to ensure uniformity and eliminate case-related discrepancies, all text is uniformly converted to lowercase before comparison. These meticulous manipulations collectively enable a fair and consistent evaluation of the accuracy of the LLM model's output concerning the labeled queries, ensuring reliable and insightful analysis of model performance.



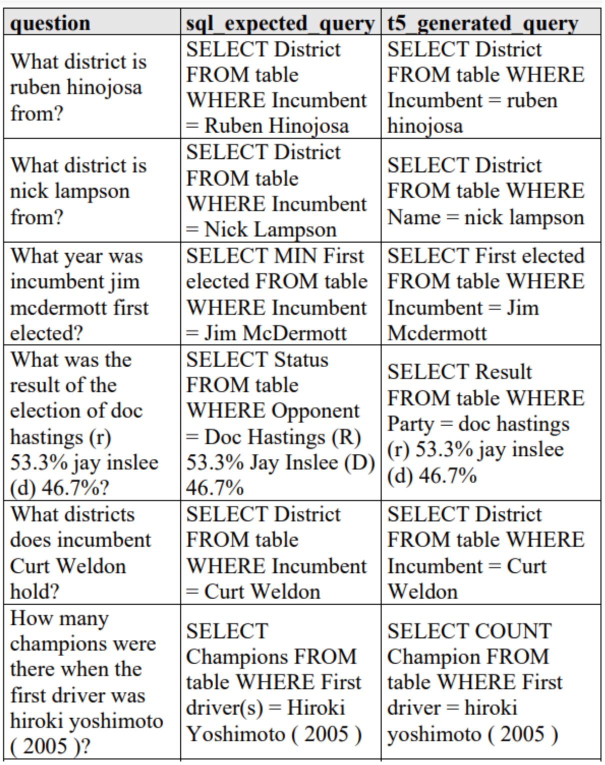
Manipulation	LLM Generated Query
LLM Generated query	"SELECT notes FROM regions WHERE region = 'South Australia';"
Replace table name with word "table"	"SELECT notes FROM table WHERE region = 'South Australia';
Remove ; from the end	"SELECT notes FROM regions WHERE region = 'South Australia'"
Convert to lower case	select notes from table where region = 'south australia'
Remove ' '	select notes from table where region = south australia

Figure 2: LLM Output Manipulation

RESULT

The project presents a comprehensive analysis of the performance of our implemented models, the T5 transformers, and the LLM model, in generating SQL queries based on the WikiSQL dataset. Through rigorous evaluation utilizing the devised metrics, including the comparison of common words between the Ground Reality Query (GRQ) and the Generated Query (GTQ), we obtained valuable insights into the models' effectiveness. Our findings reveal that both models exhibit promising capabilities in accurately generating SQL queries, with the LLM model showcasing particularly notable performance enhancements following the manipulation steps outlined in the evaluation metrics section. By standardizing the output and aligning it with the WikiSQL labeling conventions, we ensure a fair and robust comparison against the ground truth. These results not only validate the efficacy of our models but also underscore the potential of leveraging advanced natural language processing techniques for precise query generation tasks.

T5 Model Result:

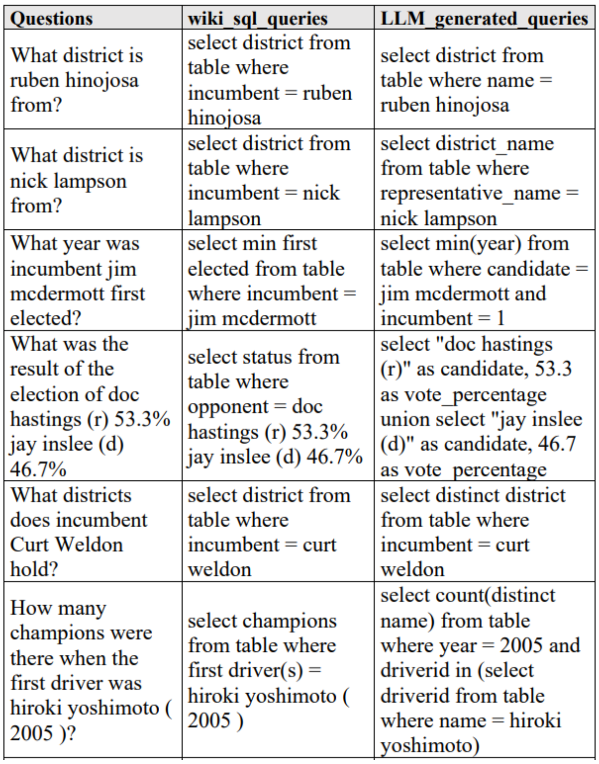


question	sql_expected_query	t5_generated_query
What district is ruben hinojosa from?	SELECT District FROM table WHERE Incumbent = Ruben Hinojosa	SELECT District FROM table WHERE Incumbent = ruben hinojosa
What district is nick lampson from?	SELECT District FROM table WHERE Incumbent = Nick Lampson	SELECT District FROM table WHERE Name = nick lampson
What year was incumbent jim mcdermott first elected?	SELECT MIN First elected FROM table WHERE Incumbent = Jim McDermott	SELECT First elected FROM table WHERE Incumbent = Jim Mcdermott
What was the result of the election of doc hastings (r) 53.3% jay inslee (d) 46.7%?	SELECT Status FROM table WHERE Opponent = Doc Hastings (R) 53.3% Jay Inslee (D) 46.7%	SELECT Result FROM table WHERE Party = doc hastings (r) 53.3% jay inslee (d) 46.7%
What districts does incumbent Curt Weldon hold?	SELECT District FROM table WHERE Incumbent = Curt Weldon	SELECT District FROM table WHERE Incumbent = Curt Weldon
How many champions were there when the first driver was hiroki yoshimoto (2005)?	SELECT Champions FROM table WHERE First driver(s) = Hiroki Yoshimoto (2005)	SELECT COUNT Champion FROM table WHERE First driver = hiroki yoshimoto (2005)

Figure 3: T5 Output

LLM Model Result:
Accuracy Comparison:

Both the T5 transformers and the LLM model have demonstrated notable performance in SQL query generation tasks based on the WikiSQL dataset. The T5 transformers yielded an accuracy of 58.06%, while the LLM model achieved a higher accuracy of 64.19%. These results underscore the effectiveness of leveraging advanced natural language processing models for generating SQL queries from natural language questions.



Questions	wiki_sql_queries	LLM_generated_queries
What district is ruben hinojosa from?	select district from table where incumbent = ruben hinojosa	select district from table where name = ruben hinojosa
What district is nick lampson from?	select district from table where incumbent = nick lampson	select district_name from table where representative_name = nick lampson
What year was incumbent jim mcdermott first elected?	select min first elected from table where incumbent = jim mcdermott	select min(year) from table where candidate = jim mcdermott and incumbent = 1
What was the result of the election of doc hastings (r) 53.3% jay inslee (d) 46.7%	select status from table where opponent = doc hastings (r) 53.3% jay inslee (d) 46.7%	select "doc hastings (r)" as candidate, 53.3 as vote_percentage union select "jay inslee (d)" as candidate, 46.7 as vote_percentage
What districts does incumbent Curt Weldon hold?	select district from table where incumbent = curt weldon	select distinct district from table where incumbent = curt weldon
How many champions were there when the first driver was hiroki yoshimoto (2005)?	select champions from table where first driver(s) = hiroki yoshimoto (2005)	select count(distinct name) from table where year = 2005 and driverid in (select driverid from table where name = hiroki yoshimoto)

Figure 4: LLM Output

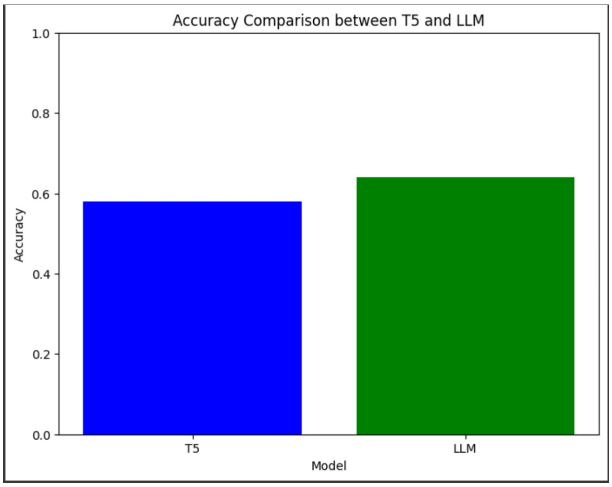


Figure 5: Accuracy Comparison

The observed performance difference between the two models suggests varying degrees of efficacy in capturing the semantics and nuances of the input questions and translating them into precise SQL queries. The higher accuracy achieved by the LLM model highlights its potential for enhancing the accuracy and efficiency of query generation systems. These findings not only validate the viability of modern NLP techniques for SQL query generation but also provide valuable insights into the comparative strengths and weaknesses of different model architectures.

In addition to the overall accuracy comparison, it is noteworthy that the Gemini API LLM model outperformed the T5 transformers across specific query operations, particularly in handling aggregation, union, and join operations. The LLM model demonstrated superior performance in accurately generating SQL queries involving these operations, showcasing its ability to capture and understand the underlying relational algebra concepts inherent in these operations. This nuanced understanding enabled the LLM model to produce more precise and contextually relevant queries for tasks requiring aggregation functions, union operations to combine multiple result sets and join operations to merge data from multiple tables based on common attributes. These results highlight the LLM model's capability to excel in complex query scenarios, further emphasizing its potential for enhancing the efficiency and accuracy of query generation systems in real-world applications.

CONCLUSION

In conclusion, our study highlights the considerable potential of leveraging state-of-the-art natural language processing models, such as the T5 transformers and the LLM model, for SQL query generation tasks based on the WikiSQL dataset. Through meticulous evaluation and analysis, we have demonstrated the effectiveness of these models in accurately generating SQL queries, as evidenced by their performance against established evaluation metrics. The manipulation steps employed to standardize the LLM model's output have yielded promising results, further enhancing its capabilities and underscoring the importance of preprocessing techniques in fine-tuning model performance. However, our work also identifies areas for future exploration and improvement. One avenue for future research involves investigating techniques to enhance the contextual understanding and reasoning capabilities of the models, potentially through the integration of external knowledge sources or fine-tuning strategies. Additionally, exploring larger and more diverse datasets could facilitate the development of more robust models capable of handling a wider range of query generation tasks across various domains.

To further advance the accuracy of natural language to SQL (NL2SQL) conversion systems, future research should prioritize several key areas. Firstly, exploring techniques to enhance the contextual understanding and reasoning capabilities of the models is paramount. By integrating external knowledge sources or developing more sophisticated fine-tuning strategies, models can better grasp the nuances of natural language queries, leading to more accurate SQL query generation.

Secondly, the exploration of larger and more diverse datasets can significantly contribute to improving accuracy. While existing datasets like WikiSQL offer comprehensive coverage, expanding the dataset repertoire to include a wider range of query scenarios and domain-specific terminologies can better reflect real-world use cases, thereby enhancing the robustness and generalizability of models.

Moreover, integrating human feedback and interactive learning mechanisms into NL2SQL conversion systems can enhance accuracy. By incorporating user feedback loops or human-in-the-loop approaches, systems can iteratively refine their performance based on real-world usage scenarios and user preferences, leading to more accurate and user-friendly interactions.

References

[1] Bu, Yanbin, et al. “A Semi-Supervised Learning Approach for Semantic Parsing Boosted by BERT Word Embedding.” Journal of Intelligent Fuzzy Systems, vol. Preprint, no. Preprint, 1 Jan. 2024, pp. 1–12,content.iospress.com/articles/journal-of-intelligent-and-fuzzy-systems/ifs233212, https://doi.org/10.3233/JIFS233212. Accessed 15 Feb. 2024.

[2] Zhong, Victor, et al. SEQ2SQL: GENERATING STRUCTURED QUERIES from NATURAL LANGUAGE USING REINFORCEMENT LEARNING. 9 Nov. 2017.

[3] Lee, Chia-Hsuan, et al. KaggleDBQA: Realistic Evaluation of Text-To-SQL Parsers.

[4] He, Yipeng, et al. Text-To-SQL Translation with Various Neural Networks CS224N Project Final Report.

[5] Kim, Hyeonji, et al. “Natural Language to SQL.” Proceedings of the VLDB Endowment, vol. 13, no. 10, June 2020, pp. 1737–1750, https://doi.org/10.14778/3401960.3401970.

[6] “Papers with Code - MAC-SQL: A Multi-Agent Collaborative Framework for Text-To-SQL”

[7] Sutskever, I., Vinyals, O., Le, Q. V. (2014). Sequence to sequence learning with neural networks. In Advances in neural information processing systems (pp. 3104-3112).

[8] Dong, L., Lapata, M. (2016). Language to logical form with neural attention. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 33-43).

[9] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Liu, P. J. (2019). Exploring the limits of transfer learning with a unified text-to-text transformer. arXiv preprint arXiv:1910.10683