OPEN ACCESS

Saudi Journal of Engineering and Technology
Abbreviated Key Title: Saudi J Eng Technol
ISSN 2415-6272 (Print) |ISSN 2415-6264 (Online)
Scholars Middle East Publishers, Dubai, United Arab Emirates
Journal homepage: https://saudijournals.com

Review Article

An Experiment on Transforming Vietnamese Natural Language Queries into SQL Statement

Khoa Dang Ho[1], Anh Hong Truong[1], Y Nhu Le[1], Khoi Minh Nguyen[1], Anh Thi-Ngoc Pham[1], Hien Tran-Hy Luong[2*]

[1]Student of Faculty of Information Technology, Ho Chi Minh City University of Education, Vietnam
[2]Lecturer of Faculty of Information Technology, Ho Chi Minh City University of Education, Vietnam

DOI: https://doi.org/10.36348/sjet.2025.v10i04.012 | **Received:** 18.03.2025 | **Accepted:** 25.04.2025 | **Published:** 29.04.2025

***Corresponding author:** Hien Tran-Hy Luong
Lecturer of Faculty of Information Technology, Ho Chi Minh City University of Education, Vietnam

Abstract

This paper explores the methodologies and results of an experiment to transform Vietnamese natural language queries into SQL statements. The paper overviews existing Text2SQL models, including state-of-the-art architectures such as T5, GPT, and BERT. These models have demonstrated the ability to transform natural language into SQL with high accuracy, but still face some challenges in handling the semantics and context of the query. This study focuses on developing an effective transformation model and analysing the unique challenges of Vietnamese, a language with a different grammatical and syntactic structure than other languages. The paper also proposes a specific transformation model, combining language preprocessing techniques, a T5-based core model, and postprocessing methods to optimise the generated SQL statements. The transformation process is detailed, from input analysis to generating the final SQL statement. Experimental results and evaluation of the test model show that the proposed model can convert Vietnamese queries to SQL with high accuracy and point out future development directions, including expanding the dataset and improving the ability to handle complex cases in the future.

Keywords: Text-to-SQL, Vietnamese Natural Language Processing, Query Transformation, SQL Generation, SQL.

1. INTRODUCTION

SQL (Structured Query Language) is the standard language for managing and querying relational databases. However, writing SQL queries can be challenging for non-technical users, especially those unfamiliar with programming. For Vietnamese users, this challenge is compounded by the lack of localized learning resources, as most SQL tutorials and documentation are available primarily in English. Addressing this gap could significantly improve database accessibility and usability for Vietnamese users.

The inherent complexity of SQL syntax – requiring precise and structured commands – poses a significant barrier to non-experts. To mitigate these challenges, this paper explores methods for automatically translating Vietnamese natural language queries into accurate SQL statements. By enabling users to interact with databases in their native language, this approach aims to democratize database access and reduce reliance on technical expertise.

The primary goal of this study is to develop a robust system capable of converting Vietnamese natural language queries into syntactically correct SQL statements. Leveraging advanced natural language processing (NLP) techniques and machine learning models, the proposed solution seeks to streamline database interactions for Vietnamese users, enhancing both efficiency and accessibility.

2. LITERATURE REVIEW

2.1 Natural Language to SQL Transformation

The transformation of natural language queries into SQL statements has been a focal point of research in NLP and database management. Early approaches relied heavily on rule-based systems, which required extensive manual effort to define transformation rules and were limited in handling complex queries. These systems often struggled with ambiguity and variability in natural language, leading to inaccurate query transformations.

Text-to-SQL (Text2SQL) refers to the automated conversion of natural language queries into executable SQL statements. Recent models typically leverage one or more of the following architectural paradigms:

a. **T5 (Text-to-Text Transfer Transformer) Based Model:** This is a model that can flexibly convert NLP tasks into text-to-text format. Especially when combined with the PICARD algorithm, it shows strong performance and gives good results on complex datasets (Rajkumar *et al*., 2022).
b. **GPT-Based Model:** The GPT-4-based model achieves high execution accuracy (up to 85.3%) on standard Text2SQL benchmarks. Using techniques like prompt decomposition and inference chaining to enhance query understanding (Hari *et al*., 2023).
c. **BERT-based model:** This model takes advantage of strong context understanding to precise query interpretation. It often fine-tuned on specialized Text2SQL datasets for optimal performance (Bhaskar *et al*., 2023).

Traditional approaches were predominantly rule-based, requiring substantial manual effort and proving inadequate in handling complex or ambiguous queries (Mohammadjafari *et al*., 2024). Recent developments have shifted toward machine learning and large language models (LLMs), which have markedly improved the accuracy and adaptability of text-to-SQL systems. According to a comprehensive review by Mohammadjafari *et al*., (2024), the field has progressed from rigid rule-based methods to advanced systems such as Retrieval-Augmented Generation (RAG), which enhance contextual understanding and schema linking while addressing computational efficiency and robustness.

Ma *et al*., (2025) proposed SQL-R1, a novel NL2SQL reasoning model trained using reinforcement learning algorithms. This model achieves competitive accuracy rates on benchmark datasets, demonstrating the potential of reinforcement learning to enhance the reasoning performance of NL2SQL models in complex scenarios involving multi-table joins and nested queries. The study emphasizes the importance of reinforcement learning in improving the adaptability and accuracy of NL2SQL systems.

2.2 Vietnamese Language NLP Research

Query analysis in Vietnamese natural language is an interesting problem that is attracting significant attention in the information technology research and development community. With the rapid development of online applications and services, the need for efficient and natural data querying is increasing.

Research specific to Vietnamese NLP has made significant strides in recent years. Vu *et al*., (2018) developed VnCoreNLP, a fast and accurate NLP annotation pipeline for Vietnamese. VnCoreNLP supports essential NLP tasks such as word segmentation, part-of-speech tagging, named entity recognition, and dependency parsing, providing state-of-the-art results. This tool has facilitated research on Vietnamese NLP by offering rich linguistic annotations and robust performance.

The GitHub repository "awsome-vietnamese-nlp" curated by vndee provides a comprehensive collection of Vietnamese NLP resources. This repository includes pre-trained language models, sentiment analysis tools, named entity recognition systems, and various datasets tailored for Vietnamese language processing. These resources are crucial for developing and fine-tuning NLP models that can accurately handle Vietnamese natural language queries.

Despite these advancements, there remains a gap in research specifically focused on transforming Vietnamese natural language queries into SQL statements. Most existing studies and models are tailored for English or other widely spoken languages and may not perform optimally with Vietnamese due to linguistic differences and the lack of extensive Vietnamese-specific training data. This paper aims to address this gap by developing a system that leverages the latest NLP techniques to transform Vietnamese natural language queries into SQL statements accurately.

3. OUR PROPOSAL MODEL

The proposed model in this study is designed to address the specific challenges Vietnamese face when converting Natural Language Queries (NLQ) to SQL statements. The model consists of three main components: language preprocessing, main model, and postprocessing, integrated into a comprehensive conversion process. Below are the details of each element.

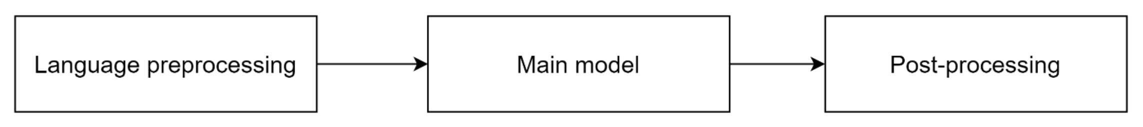

Figure 1: Our proposed model

3.1. Language Preprocessing

Preprocessing is the first step in preparing input data from Vietnamese natural language queries. Vietnamese has complex linguistic characteristics such as not using spaces to separate words, flexible sentence structure and high polysemy. Therefore, preprocessing steps include:

a. Word separation and sentence segmentation

We used VnCoreNLP to split the query into meaningful words or phrases, for example:

- Input: "*Lấy tên và tuổi của nhân viên có lương lớn hơn 10 triệu*".
- Output: *["Lấy", "tên", "và", "tuổi", "của", "nhân viên", "có", "lương", "lớn", "hơn", "10 triệu"]*.

b. Part-of-speech tagging (POS Tagging)

We used VnCoreNLP to identify the grammatical role of each word in the sentence (noun, verb, adjective, etc.), helping the model understand the semantics and structure of the sentence.

c. Named Entity Recognition (NER)

The PhoBERT is used to identify important entities such as table names, columns, specific values (e.g. "10 triệu" is a numeric value, "nhân viên" is a table).

d. Vocabulary Normalization

Synonyms or abbreviations are standardized to ensure consistency using ViSoLex (Nguyen *et al.*, 2025). For example:

“Lương” and “thu nhập” are normalized to "salary".

3.2. Main Model

The main model is the core component that performs the conversion from natural language queries to SQL statements. This model is based on the T5 (Text-to-Text Transfer Transformer) architecture, fine-tuned on the Vietnamese dataset (ViText2SQL). The main components of the model include:

a. Transformer Architecture

Transformer is a powerful architecture in natural language processing, especially tasks that require understanding the context and relationships between words in a sentence.

How it works:

- Input: Preprocessed query.
- Output: Corresponding SQL statements.

B. Fine-Tuning on Vietnamese Data

The proposed model uses the ViText2SQL dataset (~10,000 pairs of Vietnamese queries and SQL statements) to optimize the model to understand the specific semantics and syntax of Vietnamese.

c. RAG (Retrieval-Augmented Generation) Integration

Retrieve similar examples from the database to support the model in generating SQL statements to improve accuracy (Gurawa & Dharmik, 2025). For example, if the query “Lấy tên nhân viên có lương lớn hơn 10 triệu” (Get employee names with salary greater than 10 million) is similar to a known query, the model will use information from that query to improve accuracy.

d. Special Attention Mechanism

Aiming to improve the ability to focus on important keywords in queries, such as table names, columns, and filter conditions, we use attention mechanism for this case.

e. Prompt Engineering

We design prompts to guide the model to generate more accurate SQL statements. For example: Prompt: “Viết câu lệnh SQL để lấy thông tin từ bảng nhân viên với Condition lương lớn hơn 10 triệu đồng” (Write a SQL statement to get information from the employee table with the condition that salary is greater than 10 million VND).

3.3. Post-Processing

Post-processing is the final step to ensure the generated SQL statement is valid and optimize query performance. The steps include:

a. Check SQL Syntax

We used SQL parser to ensure the SQL statement is free of syntax errors before execution.

b. Query Optimization

Step by step to use optimization techniques such as adding indexes, using temporary tables, or optimizing joins. For example: Instead of using SELECT *, select only the necessary columns to reduce database load performance.

c. Feedback and Continuous Learning

The system receives feedback from the user about the accuracy of the SQL statement. The system then uses this feedback to improve the model through additional training rounds.

3.4. Overall Transformation Process

The transformation process from Vietnamese queries to SQL in the proposed model is carried out through the following steps:

- **Entering natural language queries**: The user enters a Vietnamese query, for example: “Lấy tên và tuổi của nhân viên có lương lớn hơn 10 triệu” (Get the name and age of employees with a salary greater than 10 million).
- **Preprocessing**: The query is analyzed, separated, labeled with word types, identified by entities, and normalized.
- **SQL statement generation**: The main model (T5 fine-tuned) converts the query into an SQL statement.
- **Postprocessing**: The SQL statement is checked for syntax and optimized.
- **Execution and feedback**: The SQL statement is executed on the database, and the results are returned to the user.

4. EXPERIMENTAL RESULTS AND ANALYSIS

4.1 Experimental Setup

We use ViText2SQL (Tuan Nguyen *et al*., 2020) as the main dataset. This is the first large-scale dataset for Vietnamese Text-to-SQL, including 10,000 query-SQL pairs translated from the Spider dataset, diverse in topics and query complexity, widely used in evaluating Vietnamese Text-to-SQL models. Based on suggestions from previous research (Finegan-Dollak *et al*., 2018), we split the data into training, validation, and testing datasets.

In addition, we tested our model on the Vietnamese administrative unit database (Thang Le Quoc, 2025) and MultiSpider: Multilingual dataset including Vietnamese (Dou *et al*., 2023).

4.2 Illustrative Example of the Proposed Model

To illustrate the model of transforming Vietnamese natural language queries to SQL, we will consider a specific example from input of a Vietnamese query until generating SQL statements.

Example 1: Simple Query

Vietnamese natural language query: “Lấy tên và tuổi của tất cả nhân viên trong công ty” (Get the names and ages of all employees in the company).

Transformation process:

1. Preprocessing:

- ○ **Word separation**: ["Lấy", "tên", "và", "tuổi", "của", "tất cả", "nhân viên", "trong", "công ty"]
- ○ **Part-of-Speech tagging**:
- ○ "Lấy" (verb)
- ▪ "tên" (noun)
- ▪ "tuổi" (noun)
- ▪ "nhân viên" (noun)
- ▪ "công ty" (noun)
- ○ **Named Entity Recognition**:
- ▪ Table: "nhân viên" (employees)
- ▪ Column: "tên" (name), "tuổi" (age)

2. Generate SQL statement:

The model uses T5 architecture to transform preprocessed queries into SQL statements. Generated SQL statement is:

SELECT name, age FROM employees;

Example 2: Query with Condition

Vietnamese natural language query: “Tìm tất cả sản phẩm có giá lớn hơn 100.000 đồng” (Find all products with price greater than 100,000 VND).

Transformation process:

1. Preprocessing:

- ○ **Word separation**: ["Tìm", "tất cả", "sản phẩm", "có", "giá", "lớn", "hơn", "100.000", "đồng"]
- ○ **Part-of-Speech tagging**:
  - ▪ "Tìm" (verb)
  - ▪ "sản phẩm" (noun)
  - ▪ "giá" (noun)
  - ▪ "100.000" (number)
- ○ **Named Entity Recognition**:
  - ▪ Table: "sản phẩm" (products)
  - ▪ Column: "giá" (price)
  - ▪ Condition: "lớn hơn 100.000"

2. Generate SQL statement:

Generated SQL statement is:

SELECT * FROM products WHERE price > 100000;

Example 3: Complex Query

Vietnamese natural language query: “Lấy tên sản phẩm và tổng doanh thu cho mỗi sản phẩm, chỉ tính những sản phẩm có doanh thu lớn hơn 1 triệu đồng, sắp xếp theo doanh thu giảm dần” (Get the product name and total revenue for each product, only count products with revenue greater than 1 million VND, sort by decreasing revenue).

Transformation process:

1. Preprocessing:

- ○ **Word separation**: ["Lấy", "tên", "sản phẩm", "và", "tổng", "doanh thu", "cho", "mỗi", "sản phẩm", "chỉ", "tính", "những", "sản phẩm", "có", "doanh thu", "lớn", "hơn", "1 triệu", "đồng", "sắp xếp", "theo", "doanh thu", "giảm", "dần"]
- ○ **Part-of-Speech tagging**:
  - ▪ "Lấy" (verb)
  - ▪ "tên" (noun)
  - ▪ "doanh thu" (noun)
  - ▪ "sản phẩm" (noun)
  - ▪ "1 triệu" (number)
- ○ **Named Entity Recognition**:
  - ▪ Table: "sản phẩm" (products)
  - ▪ Column: "tên", "doanh thu"
  - ▪ Condition: "doanh thu lớn hơn 1 triệu"
  - ▪ Order: "giảm dần" (DESC)

2. Generate SQL statement:

Generated SQL statement is:

SELECT product_name, SUM (revenue) AS total_revenue
FROM sales
GROUP BY product_name
HAVING total_revenue > 1000000
ORDER BY total_revenue DESC;

The above examples illustrate transforming Vietnamese natural language queries to SQL statements through the proposed model. This model not only helps users easily query data but also ensures accuracy and efficiency in processing complex queries.

4.3 Evaluation Challenges

Currently, we do not evaluate on specific datasets. We only propose a model/solution for a specific case, which is to support non-IT learners to learn database courses easily. In the future, we can record the query analysis process, SQL statement generation and query execution results. Then we will compare with other models based on benchmark dataset, analyze the advantages and disadvantages.

5. CONCLUSION

This study proposed a model for converting Vietnamese queries to SQL, taking advantage of advanced natural language processing technologies and Vietnamese-specific resources. Experimental results demonstrate the potential of the proposed method and point out future improvement directions, such as integrating large language models, expanding the ViText2SQL dataset, improving the ability to handle complex and ambiguous cases, and developing specialised evaluation tools for Vietnamese.

Acknowledgements

This research is funded by Ho Chi Minh City University of Education Foundation for Science and Technology under the student scientific research project for the academic year 2024–2025.

REFERENCES

- Anh Tuan Nguyen, Mai Hoang Dao, and Dat Quoc Nguyen (2020). *A Pilot Study of Text-to-SQL Semantic Parsing for Vietnamese*. Findings of the Association for Computational Linguistics: EMNLP 2020, pages 4079–4085. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.findings-emnlp.364
(Tuan Nguyen *et al.*, 2020)
- Bhaskar, A., Tomar, T., Sathe, A., & Sunita Sarawagi. (2023). Benchmarking and Improving Text-to-SQL Generation under Ambiguity. *ArXiv (Cornell University)*. https://doi.org/10.18653/v1/2023.emnlp-main.436
(Bhaskar *et al.*, 2023)
- Chen, W., Wang, H., Chen, J., Zhang, Y., Wang, H., Li, S., Zhou, K., & Yang, W. (2024). CoE-SQL: In-Context Learning for Multi-Turn Text-to-SQL with Chain-of-Editions. *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics*, 567-578.
- Dong, L., & Lapata, M. (2023). Next-Generation Database Interfaces: A Survey of LLM-based Text-to-SQL. *Computational Linguistics*, 49(2), 315-355.
- Dou, L., Gao, Y., Pan, M., Wang, D., Che, W., Zhan, D., & Lou, J.-G. (2023). MultiSpider: Towards Benchmarking Multilingual Text-to-SQL Semantic Parsing. *Proceedings of the AAAI Conference on Artificial Intelligence*, *37*(11), 12745–12753. https://doi.org/10.1609/aaai.v37i11.26499
(Dou *et al.*, 2023)
- Finegan-Dollak, C., Kummerfeld, J., Zhang, L., Ramanathan, K., Sadasivam, S., Zhang, R., & Radev, D. (2018). *Improving Text-to-SQL Evaluation Methodology* (pp. 351–360). https://aclanthology.org/P18-1033.pdf
(Finegan-Dollak *et al.*, 2018)
- Gurawa, P., & Dharmik, A. (2025). *Balancing Content Size in RAG-Text2SQL System*. ArXiv.org. https://arxiv.org/abs/2502.15723
(Gurawa & Dharmik, 2025)
- Hari, S., Zeng, L., & Hakkani-Tur, D. (2023). *Conversational Text-to-SQL: An Odyssey into State-of-the-Art and Challenges Ahead*. ArXiv.org. https://arxiv.org/abs/2302.11054
(Hari *et al.*, 2023)
- Ma, P., Zhuang, X., Xu, C., Jiang, X., Chen, R., & Guo, J. (2025). *SQL-R1: Training Natural Language to SQL Reasoning Model By Reinforcement Learning*. ArXiv.org. https://arxiv.org/abs/2504.08600
(Ma *et al.*, 2025)
- Mohammadjafari, A., Maida, A. S., & Gottumukkala, R. (2024). *From Natural Language to SQL: Review of LLM-based Text-to-SQL Systems*. ArXiv.org. https://arxiv.org/abs/2410.01066
(Mohammadjafari *et al.*, 2024)
- Nguyen, A. T.-H., Nguyen, D. H., & Nguyen, K. V. (2025). ViSoLex: An Open-Source Repository for Vietnamese Social Media Lexical Normalization. *ACL Anthology*, 183–188. https://aclanthology.org/2025.coling-demos.18/
(Nguyen *et al.*, 2025)
- Rajkumar, N., Li, R., & Bahdanau, D. (2022). Evaluating the Text-to-SQL Capabilities of Large Language Models. *ArXiv:2204.00498 [Cs]*. https://arxiv.org/abs/2204.00498
(Rajkumar *et al.*, 2022)
- Renggli, C., Ilyas, I. F., & Rekatsinas, T. (2025). *Fundamental Challenges in Evaluating Text2SQL Solutions and Detecting Their Limitations*. ArXiv.org. https://arxiv.org/abs/2501.18197
(Renggli *et al.*, 2025)
- Thang Le Quoc. (2025, March 12). *GitHub - ThangLeQuoc/vietnamese-provinces-database: A complete SQL dataset of Vietnamese administrative units, includes Vietnamese provinces, districts and wards*. GitHub. https://github.com/ThangLeQuoc/vietnamese-provinces-database
(Thang Le Quoc, 2025)
- vndee. (2020). *GitHub - vndee/awsome-vietnamese-nlp: A collection of Vietnamese Natural Language Processing resources*. GitHub. https://github.com/vndee/awsome-vietnamese-nlp
(vndee, 2020)