Brandenburg University of Technology

IT Security
Computerscience and Media
Prof. Dr. Oleg Lobachev
Florian Eich

Natural Language Queries using Large Language Models

Bachelor Thesis

Summer semester 2025

April 21, 2025

Mara Schulke – Matr-Nr. 20215853

Abstract

This thesis explores the integration of large language models (LLMs) into PostgreSQL database systems in order to make the database accessible via natural language instead of the postgres SQL dialect. The research focuses on implementation strategies, performance optimization, and practical applications of this concept.

Contents

List of Figures

List of Abbreviations

GPT	Generative Pretrained Transformer
SQL	Structured Query Language
API	Application Programming Interface
LLM	Large Language Model
DBMS	Database Management System
NL2SQL	Natural Language to SQL

1 Introduction

1.1 Problem Statement and Motivation

Database systems represent a backbone of modern computer science, allowing for rapid advancements whilst shielding us from the problem categories that come along with managing and querying large amounts of, usually structured, data efficiently. However, most Database Management Systems (DBMS) have traditionally required specialized knowledge, usually of the Structured Query Language (SQL), in order to become useable. Whilst this barrier may be percieved differently across diverse usergroups it represents a fundamental misalignment between end-user goals (e.g. analysts, researchers, domain experts etc.) and the underlying DBMS, thus often requiring software engineering efforts in order to reduce this friction.

This barrier is the reason entire classes of software projects exists (for example, admin / support panels), data analytics tools etc. which therefore introduce significant churn and delay between the implementation of a database system and reaching the desired end user impact. Often these projects span multiple years, require costly staffing and yield little to no novel technical value.

Emerging technologies such as Large Language Models (LLMs) have proven themselves as a sensible tool for bridging fuzzy user provided input into discrete, machine readable formats. Prominent models in this field have demostrated outstanding capabilities that enable computer scientists to tackle new problem classes, that used to be challenging / yielded unsatisfying results with discrete programming approaches.

This thesis is exploring ways to overcome the above outlined barrier using natural language queries, so that domain experts, business owners, support staff etc. are able to seamlessly interact with their data, essentially eliminating the requirement of learning SQL (and its pitfalls). By translating natural language to SQL using Large Language Models this translation becomes very robust (e.g. against different kinds of phrasing) and enables novel applications in how businesses, researchers and professionals interact with their data — it represents a fundamental shift (ie. moving away from SQL) towards a more inclusive and data driven world.

1.2 Objectives of the Thesis

This thesis aims to address the aforementioned challanges when it comes to database accessibility. The following objectives are the core research area of this thesis:

1. Develop a database extension that can translate natural language queries into semantically accurate SQL queries using Large Language Models.
2. To evaluate the effectiveness and feasibility of different Models aswell as prompt engineering techniques in order to improve the performance of the system.
3. Identify and address issues when it comes to handling amibguous, complex and domain specific user input.
4. Benchmark the performance of the implementation against common natural language to SQL (NL2SQL) benchmarks.
5. Idenitfy potential use cases for real world scenarios that could deliver a noticable upsides to users.
6. Analyze the short commings and limitations of this approach and propose potential solutions to overcome them.

1.3 Research Questions

RQ1 — Are natural language database interfaces feasible for real world application?

The primary research questions when it comes to natural language database interfaces evolve around their semantic accuracy and reliability, therefore questioning their feasibility for real world usage. LLMs have notoriously been known for their ability to hallucinate / produce false, but promising outputs. This behaviour can be especially dangerous when opting for data driven decisions that rely on false data due to a mistranslation from natural language to SQL. LLMs could cause hard to understand and debug behaviour, like false computation of distributions when the intermediate format is not being shown to the user. This thesis tries to determine whether such hallucinations could be reasonably prevented and whether the associated performance and hardware requirements are suitable for a real world deployment, outside of research situations.

Specifically the two big underlying questions are:

1. Is the semantic accuracy of natural language database interfaces high enough to yield a noticable benefit to users?
2. Is it possible to run such an interface on reasonable, mass available hardware (e.g. excluding high end research GPUs).

RQ2 — What approaches are most effective in resolving ambiguity when translating natural language queries into SQL?

To provide semantically correct results ambiguity in the user-provided natural language queries must be adequately addressed. This thesis investigates various approaches to ambiguity management and resolution. Natural language queries can demonstrate ambiguity even at low levels of complexity — e.g. there are two different types of "sales" in a database schema, and the user asks to retireve "all sales".

Such situations present the second major challenge associated with the practical implementation of natural language database interfaces. The success of this concept will significantly depend on whether suitable designs and mitigation techniques can be implemented without creating problems with regards to the aforementioned performance and hardware requirements. The research focus lies on both preventative measures through optimized pre-processing stages and prompt engineering techniques as well as reactive strategies that post process LLM output, either on the basis of further user input or context inference.

RQ3 — Which strategies are increasing semantic accuracy of queries?

In order to enhance the semantic accuracy a series of improvements may be applied to the pipeline. Potential optimizations include supplying (parts of) the schema during LLM prompting, implementation of interactive contextual reasoning through a conversational interface which would allow for user refinement, the implementation of a robust SQL parsing and validation mechanism and a hybrid approach partly relying on traditional NLP preprocessing techniques. This research will quantify semantic accuracy using popular NL2SQL benchmarks and empirically evaluate the impact each approach has on the benchmark performance. Furthermore this research will take a look at the optimal combination of the aforementioned solutions in order to develop a system that strikes the right balance between accuracy and performance.

1.4 Structure of the Thesis

This thesis is following a research and development methodology in order to implement a natural language interface for databases, in particular postgres is used.

1. **Literature Review** — An analysis of the existing research in the fields of natural language interfaces (NLI) for databases, GPU integration for acceleration of database operations, and LLM/AI Model integration within database systems. This phase establishes the theoretical foundation for this research and identifies current state-of-the-art approaches, their benefits and shortcomings.

2. **Decomposition & Requirements** — Decomposing the problem statement into its fundamentals and deriving system requirements for the design phase from it. The goal of this section is to arrive at a list of functional and non-functional requirements that must be taken into account and fulfilled by the design and implementation phases respectively.

3. **System Design** — Design of a system architecture that can utilize GPU acceleration for LLM integration from within postgres. The primary goals of the system design phase are to arrive at an architecture that yields low latency natural language processing, schema-aware SQL query generation, ambiguity detection and resolution whilst maintaining a high semantic accuracy.

4. **Implementation** — The implementation of a PostgreSQL extension according to the above system design that relies on `rust` and `pgrx`. This extension will provide a GPU accelerated framework for executing LLMs, implement a natural language to query generation pipeline that relies on the SQL schema and create database functions and operators for both query generation and execution.

5. **Evaluation and Benchmarking** — An assesment framework and benchmark that introspects the implementations performance in multiple dimensions. Namely the most relevant dimensions for this thesis are:

   (a) Semantic Accuracy — Measuring the overall accuracy of results delivered for a given natural language input.

   (b) Ambiguity Resolution Capabilities — How well the system performs when confronted with ambiguous natural language input and database schemas.

   (c) Performance Metrics — Measuring the latency, throughput and resource utilization of the implementation.

6. **Discussion** — Analysis and interpretation of the evaluation phase results against the research goals of this thesis. Evaluating the performance and accuracy results recorded during the benchmarks against the question whether real world deployments of NILs are feasible. Furthermore the effectiveness of ambiguity resolution capabilities and semantic accuracy enhancement strategies are showing a statistically significant effect.

7. **Summary and Outlook** — Summarizes the contributions, addresses limitations of this thesis and the implementation, and proposes directions for future research alongside possible applications. Primary future research topics include advanced GPU optimization techniques (e.g. further quantization), accuracy and performance impact of model fine tuning, techniques, scalability of such a system in enterprise scenarios and the evaluation of security and privacy considerations (e.g. managing access control).

2 Literature Review

Test[Izacard et al., 2022]

2.1 360 Degree vergleichbare paper etc.

3 Decomposition & Requirements

3.1 Problem Decomposition

3.2 Requirements

4 System Design

4.1 Architecture Design

4.1.1 Interface Design

4.1.2 Data Model

4.1.3 Integration into SQL

4.2 Technical Implementation Strategies

5 Implementation

5.1 Development Environment and Tools

5.2 Integration of the GPT Model

5.2.1 Model Selection and Optimization

5.2.2 API Connection or Local Embedding

5.3 Development of the PostgreSQL Extension

5.3.1 SQL Functions for GPT Interactions

5.3.2 Data Type Conversion and Processing

5.3.3 Error Handling and Logging

5.4 Optimization

5.4.1 Performance Tuning

5.4.2 Memory Usage

5.4.3 Parallelization

6 Evaluation

6.1 Test Environment and Methodology

6.2 Performance Tests

6.2.1 Latency

6.2.2 Throughput

6.2.3 Scalability

6.3 Use Cases

6.3.1 Natural Language Queries

6.3.2 Text Generation Within the Database

6.3.3 Semantic Search and Text Classification

6.4 Comparison with Alternative Approaches

7 Discussion

7.1 Interpretation of Results

7.2 Limitations of the Implementation

7.3 Ethical and Data Privacy Considerations

7.4 Potential Future Developments

8 Summary and Outlook

8.1 Summary of Results

8.2 Addressing the Research Questions

8.3 Outlook for Future Research and Development

Appendix

Installation Guide

API Documentation

Code Examples

Test Data and Results

References

[Askari et al., 2024] Askari, A., Poelitz, C., and Tang, X. (2024). Magic: Generating self-correction guideline for in-context text-to-sql.

[Chang and Fosler-Lussier, 2023] Chang, S. and Fosler-Lussier, E. (2023). How to prompt llms for text-to-sql: A study in zero-shot, single-domain, and cross-domain settings.

[Deng et al., 2020] Deng, X., Awadallah, A. H., Meek, C., Polozov, O., Sun, H., and Richardson, M. (2020). Structure-grounded pretraining for text-to-sql. *CoRR*, abs/2010.12773.

[Finegan-Dollak et al., 2018] Finegan-Dollak, C., Kummerfeld, J. K., Zhang, L., Ramanathan, K., Sadasivam, S., Zhang, R., and Radev, D. R. (2018). Improving text-to-sql evaluation methodology. *CoRR*, abs/1806.09029.

[Floratou et al., 2024] Floratou, A., Psallidas, F., Zhao, F., Deep, S., Hagleither, G., Tan, W., Cahoon, J., Alotaibi, R., Henkel, J., Singla, A., Grootel, A. V., Chow, B., Deng, K., Lin, K., Campos, M., Emani, K. V., Pandit, V., Shnayder, V., Wang, W., and Curino, C. (2024). Nl2sql is a solved problem... not! In *CIDR*.

[Gao et al., 2023a] Gao, D., Wang, H., Li, Y., Sun, X., Qian, Y., Ding, B., and Zhou, J. (2023a). Text-to-sql empowered by large language models: A benchmark evaluation.

[Gao et al., 2023b] Gao, L., Madaan, A., Zhou, S., Alon, U., Liu, P., Yang, Y., Callan, J., and Neubig, G. (2023b). Pal: program-aided language models. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

[Gao et al., 2025] Gao, Y., Liu, Y., Li, X., Shi, X., Zhu, Y., Wang, Y., Li, S., Li, W., Hong, Y., Luo, Z., Gao, J., Mou, L., and Li, Y. (2025). A preview of xiyan-sql: A multi-generator ensemble framework for text-to-sql.

[Guo et al., 2019] Guo, J., Zhan, Z., Gao, Y., Xiao, Y., Lou, J.-G., Liu, T., and Zhang, D. (2019). Towards complex text-to-SQL in cross-domain database with intermediate representation. In Korhonen, A., Traum, D., and Màrquez, L., editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4524–4535, Florence, Italy. Association for Computational Linguistics.

[Ion Androutsopoulos and Thanisch, 1995] Ion Androutsopoulos, G. D. R. and Thanisch, P. (1995). Natural language interfaces to databases - an introduction. *CoRR*, cmp-lg/9503016.

[Izacard et al., 2022] Izacard, G., Lewis, P., Lomeli, M., Hosseini, L., Petroni, F., Schick, T., Dwivedi-Yu, J., Joulin, A., Riedel, S., and Grave, E. (2022). Atlas: Few-shot learning with retrieval augmented language models.

[Lei et al., 2025] Lei, F., Chen, J., Ye, Y., Cao, R., Shin, D., Su, H., Suo, Z., Gao, H., Hu, W., Yin, P., Zhong, V., Xiong, C., Sun, R., Liu, Q., Wang, S., and Yu, T. (2025). Spider 2.0: Evaluating language models on real-world enterprise text-to-sql workflows.

[Li and Jagadish, 2014] Li, F. and Jagadish, H. V. (2014). Nalir: an interactive natural language interface for querying relational databases. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, SIGMOD '14, page 709–712, New York, NY, USA. Association for Computing Machinery.

[Li et al., 2024] Li, H., Zhang, J., Liu, H., Fan, J., Zhang, X., Zhu, J., Wei, R., Pan, H., Li, C., and Chen, H. (2024). Codes: Towards building open-source language models for text-to-sql.

[Li et al., 2023] Li, J., Hui, B., Qu, G., Yang, J., Li, B., Li, B., Wang, B., Qin, B., Cao, R., Geng, R., Huo, N., Zhou, X., Ma, C., Li, G., Chang, K. C. C., Huang, F., Cheng, R., and Li, Y. (2023). Can llm already serve as a database interface? a big bench for large-scale database grounded text-to-sqls.

[Manotas et al., 2023] Manotas, I., Popescu, O., Vo, N. P. A., and Sheinin, V. (2023). Domain adaptation of a state of the art text-to-sql model: Lessons learned and challenges found.

[Popescu et al., 2003] Popescu, A.-M., Etzioni, O., and Kautz, H. (2003). Towards a theory of natural language interfaces to databases. In *Proceedings of the 8th International Conference on Intelligent User Interfaces*, IUI '03, page 149–157, New York, NY, USA. Association for Computing Machinery.

[Pourreza et al., 2024] Pourreza, M., Li, H., Sun, R., Chung, Y., Talaei, S., Kakkar, G. T., Gan, Y., Saberi, A., Ozcan, F., and Arik, S. O. (2024). Chase-sql: Multi-path reasoning and preference optimized candidate selection in text-to-sql.

[Pourreza and Rafiei, 2023] Pourreza, M. and Rafiei, D. (2023). Din-sql: decomposed in-context learning of text-to-sql with self-correction. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

[Rahaman et al., 2024] Rahaman, A., Zheng, A., Milani, M., Chiang, F., and Pottinger, R. (2024). Evaluating sql understanding in large language models.

[Rajkumar et al., 2022] Rajkumar, N., Li, R., and Bahdanau, D. (2022). Evaluating the text-to-sql capabilities of large language models.

[Scholak et al., 2021] Scholak, T., Schucher, N., and Bahdanau, D. (2021). PICARD: parsing incrementally for constrained auto-regressive decoding from language models. *CoRR*, abs/2109.05093.

[Shen et al., 2024] Shen, Z., Vougiouklis, P., Diao, C., Vyas, K., Ji, Y., and Pan, J. Z. (2024). Improving retrieval-augmented text-to-sql with ast-based ranking and schema pruning.

[Tang and Mooney, 2001] Tang, L. R. and Mooney, R. J. (2001). Using multiple clause constructors in inductive logic programming for semantic parsing. In *Proceedings of the 12th European Conference on Machine Learning*, EMCL '01, page 466–477, Berlin, Heidelberg. Springer-Verlag.

[Xue et al., 2024] Xue, S., Jiang, C., Shi, W., Cheng, F., Chen, K., Yang, H., Zhang, Z., He, J., Zhang, H., Wei, G., Zhao, W., Zhou, F., Qi, D., Yi, H., Liu, S., and Chen, F. (2024). Db-gpt: Empowering database interactions with private large language models.

[Yaghmazadeh et al., 2017] Yaghmazadeh, N., Wang, Y., Dillig, I., and Dillig, T. (2017). Sqlizer: query synthesis from natural language. *Proc. ACM Program. Lang.*, 1(OOPSLA).

[Yu et al., 2020] Yu, T., Wu, C., Lin, X. V., Wang, B., Tan, Y. C., Yang, X., Radev, D. R., Socher, R., and Xiong, C. (2020). Grappa: Grammar-augmented pre-training for table semantic parsing. *CoRR*, abs/2009.13845.

[Yu et al., 2018] Yu, T., Zhang, R., Yang, K., Yasunaga, M., Wang, D., Li, Z., Ma, J., Li, I., Yao, Q., Roman, S., Zhang, Z., and Radev, D. R. (2018). Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task. *CoRR*, abs/1809.08887.

[Zelle and Mooney, 1996] Zelle, J. M. and Mooney, R. J. (1996). Learning to parse database queries using inductive logic programming. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence - Volume 2*, AAAI'96, page 1050–1055. AAAI Press.

[Zhang et al., 2024a] Zhang, B., Ye, Y., Du, G., Hu, X., Li, Z., Yang, S., Liu, C. H., Zhao, R., Li, Z., and Mao, H. (2024a). Benchmarking the text-to-sql capability of large language models: A comprehensive evaluation.

[Zhang et al., 2024b] Zhang, H., Cao, R., Xu, H., Chen, L., and Yu, K. (2024b). Coe-sql: In-context learning for multi-turn text-to-sql with chain-of-editions.

[Zhong et al., 2017] Zhong, V., Xiong, C., and Socher, R. (2017). Seq2sql: Generating structured queries from natural language using reinforcement learning. *CoRR*, abs/1709.00103.