Brandenburg University of Applied Sciences

IT Security
Computerscience
Prof. Dr. Oleg Lobachev
MSc. Florian Eich

Reliable Natural Language Interfaces using LLMs, Self-Correction and Incremental Schema Analysis

Bachelor Thesis

Summer semester 2025

April 25, 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

In this section a comprehensive literature review is performed to asses the research landscape on NL2SQL (sometimes also referred to as Text-to-SQL or T2SQL) and NLIDBs. Following their development starting in the late 1990s and early 2000s (Androutsopoulos, Ritchie, & Thanisch, 1995; Popescu, Etzioni, & Kautz, 2003; Tang & Mooney, 2001; Zelle & Mooney, 1996) until now, observing multiple larger paradigm shifts happening over time (Deng et al., 2020; F. Li & Jagadish, 2014; Yaghmazadeh, Wang, Dillig, & Dillig, 2017; Yu et al., 2020; Zhong, Xiong, & Socher, 2017). In particular this research focuses on the recent advancements when it comes to language models and how they can be harnessed for effective NL2SQL systems (D. Gao et al., 2023; Lei et al., 2025; J. Li et al., 2023; Rahaman, Zheng, Milani, Chiang, & Pottinger, 2024; Rajkumar, Li, & Bahdanau, 2022; B. Zhang et al., 2024).

This literature review is covering the foundational concepts, challenges, key advancements and research gaps associated with using natural language instead of SQL. It lays the foundation for this thesis and helps to set the research questions introduced in the previous chapter in context.

2.1 Foundations of Natural Language Interfaces to Databases

One of the first corner stone research papers on Natural Language Database Interfaces (NLIDBs) was published over three decades ago by Androutsopoulos, Ritchie, and Thanisch where an introduction and an overview of the state-of-the-art in the field were provided. (Androutsopoulos et al., 1995) Their work outlined multiple key issues and challenges associated with NLIDBs, and compared them against existing / competing solutions like formal query languages, form-based interfaces and graphical interfaces. These challenges (like unobvious limits, linguistic ambiguities, semantic inaccuracy, tedious configuration etc.) have shaped this field of research and are still considered relevant metrics today.

Early NLIDBs primarily relied on traditional natural language processing (NLP) techniques in order to achieve natural language understanding capabilities. With Chill an inductive logic programming (ILP) approach was first introduced for NL2SQL systems, marking one of the key events when it comes to machine learning usage. (Zelle & Mooney, 1996) In 2001 Tang and Mooney have extended the approach of ILP parsing for natural language database queries with multi clause construction, yielding promising results in the field of NLIDBs. (Tang & Mooney, 2001)

Building on the systematic overview of Androutsopoulos, Ritchie, and Thanisch and the first machine learning approaches from Zelle and Mooney aswell as Tang and Mooney, Popescu et al. have proposed a novel approach for implementing NLIDBs and outperformed at the time state-of-the-art solutions from Zelle and Mooney (1996) Tang and Mooney (2001) — achieving 80% semantic accuracy. (Popescu et al., 2003) The novelty of the Percise system lies in its natural language processing approach, specifically its lexical mapping strategy, allowing Percise to identify questions it can, and can't answer (introducing the concept of *semantically tractable questions*) which therefore results in a better and interactive end user experience. Their experiements also showed that this approach is *transferrable* and *unbiased* — it is possible to feed in new, unknown questions into the system and maintain performance characteristics, whereas it was shown that Zelle and Mooney (1996) were suffering from a distribution drift of the questions asked. (Popescu et al., 2003)

The theoretical foundations and research questions highlighted by the aforementioned works, shaped the research field and highlighted the following, ongoing research:

1. The trade-off characteristics derived from choosing a machine learning vs. traditional NLP

approach (e.g. Chill versus Percise). E.g. coverage versus correctness. (Popescu et al., 2003; Zelle & Mooney, 1996)

2. The linguistic challenges associated with bringing NLIDBs into use (e.g. semantic inaccuracy, linguistic ambiguity, unclear language coverage etc.) (Androutsopoulos et al., 1995)
3. The value of systems and approaches which double down on reliability and semantic accuracy rather than giving promising but incorrect answers. (Androutsopoulos et al., 1995; Popescu et al., 2003)

Fundamentally this highlights the tension and mismatch between the characteristics of natural language, which is able to be ambiguous, *semantically untractable* or able to be incomplete in meaning and formal languages like SQL which always have on deterministic and *semantically tractable* meaning they convey in each statement. As Schneiderman and Norman have pointed out according to Popescu, Etzioni, and Kautz, users are "unwilling to trade reliable and predictable user interfaces for intelligent but unreliable ones" which induces performance expectations on NLIDB implementations to be highly certain about the questions it can, and can't answer, whilst maintaing as high as possible natural language coverage. (Popescu et al., 2003)

2.2 Traditional NL2SQL Approaches

Prior to the wide-spread dominance of machine learning approaches for natural language processing a variety of traditional, rather discrtete approaches have been explored in the field of NL2SQL / NLIDBs. These discrete approaches have laid the foundations for transitioning towards the application of machine learning techniques for NL2SQL.

2.2.1 Rule-based and Grammar-based Systems

Early research of NL2SQL system mostly focused around applying rule engines that were tedious to set up and expensive to maintain / transfer across database systems. These rule engines mostly relied on the systematic identification of linguistic patterns / were trying to template SQL from information that was derived from processing the natural language query. (Codd, 1974; Hendrix, Sacerdoti, Sagalowicz, & Slocum, 1978; Woods, Kaplan, & Nash-Webber, 1972) These approaches mostly tried to formalize natural language queries into formal grammers which could then be deterministically mapped into a valid SQL query. These approaches have strong downsides when it comes to the variety of natural language constructs they can process, aswell as runtime adoption of new / unknown databases, query constructs etc. A potential upside of this class of NL2SQL systems is that they can confidently and reproducably identify questions they can, and can't answer — thus leading to very reliable and predictable user interfaces.

2.3 Neural NL2SQL Approaches

2.4 Benchmark Evolution

2.5 LLM-based NL2SQL Systems

2.6 Optimization for LLM-based NL2SQL

2.7 Complex Techniques in NL2SQL

2.8 Research Gaps

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 Model

5.3 Development of the PostgreSQL Extension

5.4 Optimization

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

Androutsopoulos, I., Ritchie, G. D., & Thanisch, P. (1995). Natural language interfaces to databases - an introduction. *CoRR*, *cmp-lg/9503016*. Retrieved from `http://arxiv.org/abs/cmp-lg/9503016`

Askari, A., Poelitz, C., & Tang, X. (2024). *Magic: Generating self-correction guideline for in-context text-to-sql.* Retrieved from `https://arxiv.org/abs/2406.12692`

Chang, S., & Fosler-Lussier, E. (2023). *How to prompt llms for text-to-sql: A study in zero-shot, single-domain, and cross-domain settings.* Retrieved from `https://arxiv.org/abs/2305.11853`

Codd, E. F. (1974, January). Seven steps to rendezvous with the casual user. In J. W. Klimbie & K. L. Koffeman (Eds.), *Ifip working conference data base management* (p. 179-200). North-Holland. Retrieved from `http://dblp.uni-trier.de/db/conf/ds/dbm74.html#Codd74` (IBM Research Report RJ 1333, San Jose, California)

Deng, X., Awadallah, A. H., Meek, C., Polozov, O., Sun, H., & Richardson, M. (2020). Structure-grounded pretraining for text-to-sql. *CoRR*, *abs/2010.12773*. Retrieved from `https://arxiv.org/abs/2010.12773`

Finegan-Dollak, C., Kummerfeld, J. K., Zhang, L., Ramanathan, K., Sadasivam, S., Zhang, R., & Radev, D. R. (2018). Improving text-to-sql evaluation methodology. *CoRR*, *abs/1806.09029*. Retrieved from `http://arxiv.org/abs/1806.09029`

Floratou, A., Psallidas, F., Zhao, F., Deep, S., Hagleither, G., Tan, W., … Curino, C. (2024). Nl2sql is a solved problem... not! In *Cidr.* Retrieved from `https://www.cidrdb.org/cidr2024/papers/p74-floratou.pdf`

Gao, D., Wang, H., Li, Y., Sun, X., Qian, Y., Ding, B., & Zhou, J. (2023). *Text-to-sql empowered by large language models: A benchmark evaluation.* Retrieved from `https://arxiv.org/abs/2308.15363`

Gao, L., Madaan, A., Zhou, S., Alon, U., Liu, P., Yang, Y., … Neubig, G. (2023). Pal: program-aided language models. In *Proceedings of the 40th international conference on machine learning.* JMLR.org.

Gao, Y., Liu, Y., Li, X., Shi, X., Zhu, Y., Wang, Y., … Li, Y. (2025). *A preview of xiyan-sql: A multi-generator ensemble framework for text-to-sql.* Retrieved from `https://arxiv.org/abs/2411.08599`

Guo, J., Zhan, Z., Gao, Y., Xiao, Y., Lou, J.-G., Liu, T., & Zhang, D. (2019, July). Towards complex text-to-SQL in cross-domain database with intermediate representation. In A. Korhonen, D. Traum, & L. Màrquez (Eds.), *Proceedings of the 57th annual meeting of the association for computational linguistics* (pp. 4524–4535). Florence, Italy: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/P19-1444/` doi: 10.18653/v1/P19-1444

Hendrix, G. G., Sacerdoti, E. D., Sagalowicz, D., & Slocum, J. (1978, June). Developing a natural language interface to complex data. *ACM Trans. Database Syst.*, *3*(2), 105–147. Retrieved from `https://doi.org/10.1145/320251.320253` doi: 10.1145/320251.320253

Izacard, G., Lewis, P., Lomeli, M., Hosseini, L., Petroni, F., Schick, T., … Grave, E. (2022). *Atlas: Few-shot learning with retrieval augmented language models.* Retrieved from https://arxiv.org/abs/2208.03299

Kate, R. J., & Mooney, R. J. (2006, July). Using string-kernels for learning semantic parsers. In N. Calzolari, C. Cardie, & P. Isabelle (Eds.), *Proceedings of the 21st international conference on computational linguistics and 44th annual meeting of the association for computational linguistics* (pp. 913–920). Sydney, Australia: Association for Computational Linguistics. Retrieved from https://aclanthology.org/P06-1115/ doi: 10.3115/1220175.1220290

Lei, F., Chen, J., Ye, Y., Cao, R., Shin, D., Su, H., … Yu, T. (2025). *Spider 2.0: Evaluating language models on real-world enterprise text-to-sql workflows.* Retrieved from https://arxiv.org/abs/2411.07763

Li, F., & 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* (p. 709–712). New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/2588555.2594519 doi: 10.1145/2588555.2594519

Li, H., Zhang, J., Liu, H., Fan, J., Zhang, X., Zhu, J., … Chen, H. (2024). *Codes: Towards building open-source language models for text-to-sql.* Retrieved from https://arxiv.org/abs/2402.16347

Li, J., Hui, B., Qu, G., Yang, J., Li, B., Li, B., … Li, Y. (2023). *Can llm already serve as a database interface? a big bench for large-scale database grounded text-to-sqls.* Retrieved from https://arxiv.org/abs/2305.03111

Manotas, I., Popescu, O., Vo, N. P. A., & Sheinin, V. (2023). *Domain adaptation of a state of the art text-to-sql model: Lessons learned and challenges found.* Retrieved from https://arxiv.org/abs/2312.05448

Popescu, A.-M., Etzioni, O., & Kautz, H. (2003). Towards a theory of natural language interfaces to databases. In *Proceedings of the 8th international conference on intelligent user interfaces* (p. 149–157). New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/604045.604070 doi: 10.1145/604045.604070

Pourreza, M., Li, H., Sun, R., Chung, Y., Talaei, S., Kakkar, G. T., … Arik, S. O. (2024). *Chase-sql: Multi-path reasoning and preference optimized candidate selection in text-to-sql.* Retrieved from https://arxiv.org/abs/2410.01943

Pourreza, M., & 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.* Red Hook, NY, USA: Curran Associates Inc.

Rahaman, A., Zheng, A., Milani, M., Chiang, F., & Pottinger, R. (2024). *Evaluating sql understanding in large language models.* Retrieved from https://arxiv.org/abs/2410.10680

Rajkumar, N., Li, R., & Bahdanau, D. (2022). *Evaluating the text-to-sql capabilities of large language models.* Retrieved from https://arxiv.org/abs/2204.00498

Reddy, S., Lapata, M., & Steedman, M. (2014). Large-scale semantic parsing without question-answer pairs. *Transactions of the Association for Computational Linguistics*, *2*, 377–392. Retrieved from https://aclanthology.org/Q14-1030/ doi: 10.1162/tacl_a_00190

Scholak, T., Schucher, N., & Bahdanau, D. (2021). PICARD: parsing incrementally for constrained auto-regressive decoding from language models. *CoRR*, *abs/2109.05093*. Retrieved from https://arxiv.org/abs/2109.05093

Shen, Z., Vougiouklis, P., Diao, C., Vyas, K., Ji, Y., & Pan, J. Z. (2024). *Improving retrieval-augmented text-to-sql with ast-based ranking and schema pruning.* Retrieved from https://

arxiv.org/abs/2407.03227

Tang, L. R., & 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* (p. 466–477). Berlin, Heidelberg: Springer-Verlag.

Woods, W. A., Kaplan, R., & Nash-Webber, B. (1972). *The lunar sciences natural language information system: Final report.* Cambridge, Massachusetts: Bolt, Beranek and Newman, Inc.

Xue, S., Jiang, C., Shi, W., Cheng, F., Chen, K., Yang, H., ... Chen, F. (2024). *Db-gpt: Empowering database interactions with private large language models.* Retrieved from https://arxiv.org/abs/2312.17449

Yaghmazadeh, N., Wang, Y., Dillig, I., & Dillig, T. (2017, October). Sqlizer: query synthesis from natural language. *Proc. ACM Program. Lang.*, *1*(OOPSLA). Retrieved from https://doi.org/10.1145/3133887 doi: 10.1145/3133887

Yu, T., Wu, C., Lin, X. V., Wang, B., Tan, Y. C., Yang, X., ... Xiong, C. (2020). Grappa: Grammar-augmented pre-training for table semantic parsing. *CoRR*, *abs/2009.13845*. Retrieved from https://arxiv.org/abs/2009.13845

Yu, T., Zhang, R., Yang, K., Yasunaga, M., Wang, D., Li, Z., ... 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*. Retrieved from http://arxiv.org/abs/1809.08887

Zelle, J. M., & 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* (p. 1050–1055). AAAI Press.

Zhang, B., Ye, Y., Du, G., Hu, X., Li, Z., Yang, S., ... Mao, H. (2024). *Benchmarking the text-to-sql capability of large language models: A comprehensive evaluation.* Retrieved from https://arxiv.org/abs/2403.02951

Zhang, H., Cao, R., Xu, H., Chen, L., & Yu, K. (2024). *Coe-sql: In-context learning for multi-turn text-to-sql with chain-of-editions.* Retrieved from https://arxiv.org/abs/2405.02712

Zhong, V., Xiong, C., & Socher, R. (2017). Seq2sql: Generating structured queries from natural language using reinforcement learning. *CoRR*, *abs/1709.00103*. Retrieved from http://arxiv.org/abs/1709.00103