

Brandenburg University of Technology

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

Embedding Generative Pretrained Transformers into PostgreSQL

Bachelor Thesis

Summer semester 2025

March 22, 2025

Mara Schulke – Matr-Nr. 20215853

Abstract

This thesis explores the integration of Generative Pretrained Transformers (GPT) into PostgreSQL database systems. The research focuses on implementation strategies, performance optimization, and practical applications of this integration.

Contents

1	Introduction	3
1.1	Problem Statement and Motivation	3
1.2	Objectives of the Thesis	3
1.3	Research Questions	3
1.4	Methodological Approach	3
1.5	Structure of the Thesis	3
2	Theoretical Foundations	4
2.1	Generative Pretrained Transformers (GPT)	4
2.1.1	Architecture and Functionality	4
2.1.2	Training and Fine-tuning	4
2.1.3	Application Areas	4
2.2	PostgreSQL as a Database System	4
2.2.1	Architecture of PostgreSQL	4
2.2.2	Extension Capabilities	4
2.2.3	PostGIS and Other Extensions as Examples	4
2.3	Embedding AI Models in Database Systems	4
2.3.1	Current Approaches and Solutions	4
2.3.2	Technical Challenges	4
2.3.3	Benefits of Integration	4
3	Conceptual Design of GPT Embedding in PostgreSQL	5
3.1	Requirements Analysis	5
3.1.1	Functional Requirements	5
3.1.2	Non-functional Requirements	5
3.2	Architecture Design	5
3.2.1	Interface Design	5
3.2.2	Data Model	5
3.2.3	Integration into PostgreSQL	5
3.3	Technical Implementation Strategies	5
3.3.1	Foreign Data Wrapper	5
3.3.2	Extension Using C/C++	5
3.3.3	PL/Python or Other Procedural Languages	5
3.3.4	Comparison of Approaches	5
4	Implementation	6
4.1	Development Environment and Tools	6
4.2	Integration of the GPT Model	6
4.2.1	Model Selection and Optimization	6
4.2.2	API Connection or Local Embedding	6
4.3	Development of the PostgreSQL Extension	6

4.3.1	SQL Functions for GPT Interactions	6
4.3.2	Data Type Conversion and Processing	6
4.3.3	Error Handling and Logging	6
4.4	Optimization	6
4.4.1	Performance Tuning	6
4.4.2	Memory Usage	6
4.4.3	Parallelization	6
5	Evaluation	7
5.1	Test Environment and Methodology	7
5.2	Performance Tests	7
5.2.1	Latency	7
5.2.2	Throughput	7
5.2.3	Scalability	7
5.3	Use Cases	7
5.3.1	Natural Language Queries	7
5.3.2	Text Generation Within the Database	7
5.3.3	Semantic Search and Text Classification	7
5.4	Comparison with Alternative Approaches	7
6	Discussion	8
6.1	Interpretation of Results	8
6.2	Limitations of the Implementation	8
6.3	Ethical and Data Privacy Considerations	8
6.4	Potential Future Developments	8
7	Summary and Outlook	9
7.1	Summary of Results	9
7.2	Addressing the Research Questions	9
7.3	Outlook for Future Research and Development	9

List of Figures

List of Abbreviations

GPT	Generative Pretrained Transformer
SQL	Structured Query Language
API	Application Programming Interface

1 Introduction

1.1 Problem Statement and Motivation

1.2 Objectives of the Thesis

1.3 Research Questions

1.4 Methodological Approach

1.5 Structure of the Thesis

2 Theoretical Foundations

2.1 Generative Pretrained Transformers (GPT)

2.1.1 Architecture and Functionality

2.1.2 Training and Fine-tuning

2.1.3 Application Areas

2.2 PostgreSQL as a Database System

2.2.1 Architecture of PostgreSQL

2.2.2 Extension Capabilities

2.2.3 PostGIS and Other Extensions as Examples

2.3 Embedding AI Models in Database Systems

2.3.1 Current Approaches and Solutions

2.3.2 Technical Challenges

2.3.3 Benefits of Integration

3 Conceptual Design of GPT Embedding in PostgreSQL

3.1 Requirements Analysis

3.1.1 Functional Requirements

3.1.2 Non-functional Requirements

3.2 Architecture Design

3.2.1 Interface Design

3.2.2 Data Model

3.2.3 Integration into PostgreSQL

3.3 Technical Implementation Strategies

3.3.1 Foreign Data Wrapper

3.3.2 Extension Using C/C++

3.3.3 PL/Python or Other Procedural Languages

3.3.4 Comparison of Approaches

4 Implementation

4.1 Development Environment and Tools

4.2 Integration of the GPT Model

4.2.1 Model Selection and Optimization

4.2.2 API Connection or Local Embedding

4.3 Development of the PostgreSQL Extension

4.3.1 SQL Functions for GPT Interactions

4.3.2 Data Type Conversion and Processing

4.3.3 Error Handling and Logging

4.4 Optimization

4.4.1 Performance Tuning

4.4.2 Memory Usage

4.4.3 Parallelization

5 Evaluation

5.1 Test Environment and Methodology

5.2 Performance Tests

5.2.1 Latency

5.2.2 Throughput

5.2.3 Scalability

5.3 Use Cases

5.3.1 Natural Language Queries

5.3.2 Text Generation Within the Database

5.3.3 Semantic Search and Text Classification

5.4 Comparison with Alternative Approaches

6 Discussion

6.1 Interpretation of Results

6.2 Limitations of the Implementation

6.3 Ethical and Data Privacy Considerations

6.4 Potential Future Developments

7 Summary and Outlook

7.1 Summary of Results

7.2 Addressing the Research Questions

7.3 Outlook for Future Research and Development

References

- [1] Author, A. (Year). Title of the reference. Journal/Publisher, Volume(Issue), Pages.

Appendix

Installation Guide

API Documentation

Code Examples

Test Data and Results