Text-to-SQL für DataGovernance Technologies und die Aus- und Weiterbildung

Student

Fiona Pichler

Introduction: This term project was a teamwork with the bachelor thesis of Benjamin Kern. Providing non-technical professionals with actionable insights derived from data is the vision of Swiss software company DataGovernance Technologies Ltd (DGT). The emergence of Large Language Models (LLMs) has made this vision increasingly achievable, in particular by enabling the generation of SQL queries directly from natural language (NL) input. Similarly, OST seeks to address a related challenge by making the process of learning SQL more accessible and engaging for its students. Through interactive and efficient methods, OST aims to simplify the complexity of SQL for learners. This project explores different approaches to translating NL into SQL and implements a proof of concept (POC) to evaluate their potential and effectiveness using two example datasets.

Approach: The project began with an investigation of suitable methods for incorporating schema information into LLMs and existing work on the subject. This research identified four core approaches: pure LLM as a baseline, in-context learning, fine-tuning of the LLM, and Retrieval-Augmented Generation (RAG). Fine-tuning was ruled out due to an insufficient amount of training data. The remaining three approaches were implemented in Python and relevant LLM APIs and thoroughly evaluated. A number of LLMs were selected for performance evaluation, including Mistral, phi3:3.8b, phi3:14b, llama3.1:8b, and GPT-4o-mini. Two datasets were selected: One was provided by DGT (MS SQL Server) and for the OST use case it's the well-known Pagila dataset (PostgreSQL). The test SQL queries and their corresponding user prompts were extracted from database lecture slides and others were generated using ChatGPT. They were divided into basic and advanced test cases.

Result: The results of the test queries were evaluated in terms of 1. similarity (how similar is an output compared to the example solution?), 2. validity (is the output valid SQL?), 3. executability (can the output be executed and are the generated column names correct?), 4. reliability (how similar is the output to the same user prompt?). It proved difficult to evaluate similarity using the usual cosine function, as the same SQL resultset can be achieved with different SQL queries. Since the pure LLM approach guesses table names, the similarity metric was still considered useful for comparison. While executability includes validity, the validity metric can be achieved without the correct table names, which is needed for comparison with the pure LLM approach. Fig. 2 shows the result for RAG for executing the SQL queries on the databases. For the advanced test cases, on average 45% can be executed on the databases, with only 70% of the requested columns extracted. Getting the LLMs to output valid SQL without additional explanation was a challenge. The hallucinations could not be completely eliminated, resulting in the low number of executable SQL queries. The reliability metric in Fig. 3 shows the high number of hallucinations by phi3:3.8b. In this thesis, the llama3.2 model was found to have the most potential for further development. Further work could focus on eliminating the hallucinations as well as providing more fine-grained test metrics to thoroughly analyse the shortcomings of the LLM. Emerging approaches not covered by this work could be investigated, such as structured output, multi-step reasoning, and agent-like function calling.

Fig. 1: A demonstration of how the chatbot transforms a user prompt into valid SQL and executes it on a database.
Own presentment

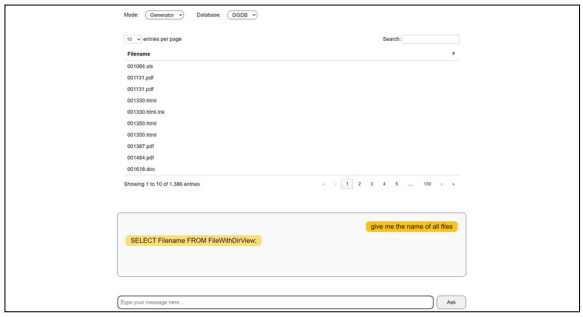

Fig. 2: Test results using RAG on the Pagila database with different LLMs, each computed over 10 test runs.
Own presentment

LLM	columns correctly identified (%)				queries executable on database (%)			
	basic testcases		advanced testcases		basic testcases		advanced testcases	
	avg	max	avg	max	avg	max	avg	max
Mistral-7b-v.01	0.66	0.73	0.51	0.88	0.25	0.30	0.06	0.09
phi3:3.8	0.56	0.78	0.43	0.7	0.40	0.45	0.08	0.09
llama3.2	0.68	0.68	0.45	0.61	0.62	0.64	0.30	0.38
phi3:14	0.62	0.69	0.42	0.5	0.17	0.21	0.12	0.19
gpt-4o-mini:	0.70	0.73	0.50	0.55	0.62	0.66	0.45	0.55

Fig. 3: Reliability calculated over 10 runs for all test cases, approaches and databases. Summarized as average per LLM.
Own presentment

LLM	Reliability (%)	Rank (1 = best)
Mistral-7b-v.01	0.86	4
phi3:3.8	0.26	5
llama3.2	0.97	1
phi3:14	0.91	3
gpt-4o-mini:	0.93	2

Advisor
Prof. Stefan F. Keller

Subject Area
Artificial Intelligence, Software