ANALYSIS ON DB-GPT: EMPOWERING DATABASE RECIPROCATION WITHPRIVATE LARGE LANGUAGE MODELS

Prof. Mandan Mishra
Assistant Professor
Sandip University Madhubani

Abstract

The recent breakthroughs in large language models (LLMs) are positioned to transition many areas of software. Database technologies particularly have an important entanglement with LLMs as efficient and intuitive database interactions are paramount. In this paper, we present DB-GPT, a revolutionary and production-ready project that integrates LLMs with traditional database systems to enhance user experience and accessibility. DB-GPT is designed to understand natural language queries, provide context-aware responses, and generate complex SQL queries with high accuracy, making it an indispensable tool for users ranging from novice to expert. The core innovation in DB-GPT lies in its private LLM technology, which is fine-tuned on domain-specific corpora to maintain user privacy and ensure data security while offering the benefits of state-of-the-art LLMs. We detail the architecture of DB-GPT, which includes a novel retrieval augmented generation (RAG) knowledge system, an adaptive learning mechanism to continuously improve performance based on user feedback and a service-oriented multi-model framework (SMMF) with powerful data-driven agents. Our extensive experiments and user studies confirm that DB-GPT represents a paradigm shift in database interactions, offering a more natural, efficient, and secure way to engage with data repositories. The paper concludes with a discussion of the implications of DB-GPT framework on the future of human-database interaction and outlines potential avenues for further enhancements and applications in the field.

1. INTRODUCTION

Large language models (LLMs) such as ChatGPT (Brown et al., 2020) and GPT-4 (OpenAI, 2023)have showcased their remarkable capabilities in engaging in human-like communication and under-standing complex queries, bringing a trend of incorporating LLMs in various fields (Anil et al., 2023;Gunasekar et al., 2023). These models have been further enhanced by external tools, enabling them to search for relevant online information (Nakano et al., 2021; Xue et al., 2023c), utilize tools (Schick et al., 2023), and create more sophisticated applications (Chase, 2022; Wang et al., 2023; Chu et al.,

	LangChain (Chase, 2022)	LlmaIndex (Liu, 2022)	PrivateGPT (Martínez et al., 2023)	ChatDB (Hu et al., 2023)	DB-GPT
Multi-LLM integration	✓	✓	✗	✓	✓
Text-to-SQL fine-tuned	✗	✓	✗	✗	✓
Multi-agent strategies	✓	✓	✗	✗	✓
Data privacy and security	✓	✗	✓	✗	✓
Multi-source knowledge	✓	✓	✗	✗	✓
Bilingual queries	✗	✗	✗	✓	✓
Generative data analytics	✗	✗	✗	✗	✓

Table 1: Comparative summary of competing approaches on various dimensions.

In the realm of databases, while traditional systems often demand a high degree of technicalacumen and familiarity with domain-specific structural query languages (SQLs) for data access and manipulation, LLMs pave the way for natural language interfaces, enabling users to express through natural language queries and leading to more natural and intuitive database interactions.

Nonetheless, how to empower the database operations with LLMs to build powerful end-user applications still remains an open question. One straightforward approach, employed by most of existing works (Chase, 2022; Zhou et al., 2023; Hu et al., 2023), is to directly providing commonly used LLMs, such as GPT-4, with instructions on how to interact via few-short prompting or in-context learning (Wei et al., 2022). The advantages of this approach is, it is unlikely to over-fit to train data and is easy to adapt to new data while the disadvantages are the performance can be sub-optimal compared to the fine-tuned alternatives with median-sized LLMs (Sun et al., 2023). Moreover, to further facilitate the intelligent interactions with database, many works (Chase, 2022; Liu, 2022; Richards, 2022) have incorporated the LLM-powered automated reasoning and decision process (a.k.a., agent) into the database applications. However, the knowledge agents are usually task-specific instead of task agnostic, limiting their use to a large scale. Meanwhile, though being important, the privacy-sensitive setup for LLM-centric database interactions have been under-investigated. The previous efforts (Martínez et al., 2023; H2O.ai, 2023) are mostly general-purpose and not specially designed for database operations.

In this work, we introduce DB-GPT, an intelligent and production-ready project for LLM-augmented applications to ingest, structure, and access data with privatization technologies. DB-GPT harnesses not only the inherent natural language understanding and generation capabilities of LLMs but also continuously optimizes the data-driven engine through the agent and plugin mechanism. See Table 1 for a comparative summary of competitors. To summarize, DB-GPT has the following distinct merits:

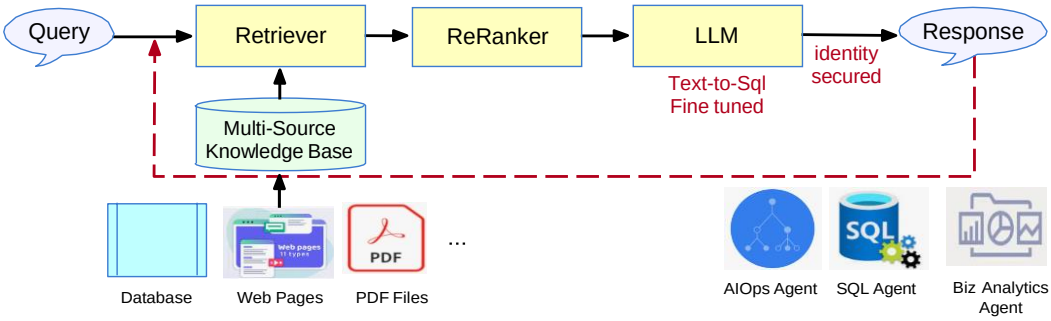

Figure 1: The architecture of DB-GPT

- **Privacy and security protection.** DB-GPT allows users to deploy on personal devices or local servers and run even in scenarios without Internet connection. No data leaves the execution environment at any point, completely eliminating the risk of data leakage. In addition, proxy de-identification (Wang et al., 2016) techniques are applied in data processing modules, which actsas an intermediary that obscures personal identifiers from datasets, thereby mitigating the risks of unauthorized access and exploitation of private information.

- **Multi-source knowledge base question & answering optimized.** In contrast to classical works (Lan et al., 2022) of knowledge base question & answering (KBQA), DB-GPT builds a pipeline that ingests multi-source unstructured data (PDF's, web pages, images, etc) into intermediate representations, stores them in a structured knowledge base, retrieves the most relevant pieces, and generates a comprehensive natural language response given a query. The pipeline is efficiency-optimized, flexible in generation and accepts bilingual queries.
- **Text-to-SQL fine-tuned.** To further enhance the generation capability, DB-GPT fine-tuned several commonly used LLMs (e.g., Llama-2 (Touvron et al., 2023), GLM (Zeng et al., 2022)) for Text-to-SQL tasks. DB-GPT significantly lowers the barriers to users without the expertise of SQL when interacting with data. To the best of our knowledge, among related works, only LlamaIndex (Liu, 2022) integrates such fine-tuned alternatives but it is not optimized for bilingual queries.
- **Knowledge agents and plugins integrated.** An "agent" is an automated reasoning and decision engine. As a production-ready project, DB-GPT enables the development and application of conversational agents with advanced data analytics, where these automated decisions help interactive use cases over the data. It also offers a variety of plugins of query and retrieval services to use as tools for interaction with data.

We rigorously evaluate DB-GPT on various benchmark tasks, such as Text-to-SQL and KBQA. Furthermore, we conduct case studies and surveys to assess the usability and preferences. DB-GPT outperforms the competitors for most of the dimensions.

2. SYSTEM DESIGN

The overall pipeline of DB-GPT is depicted in Figure 1. While building upon the general Retrieval-Augmented Generation (RAG) framework (Chase, 2022; Liu, 2022; Xue et al., 2023c), our DB-GPT system integrates our novel training and inference techniques, which significantly enhance its overall performance and efficiency. In this section, we delineate the design of each phase, including the model architecture as well as training and inference paradigms.

Multi-source RAG for QA

While LLMs are usually trained on enormous bodies of open sourced or other parties' proprietary data, RAG (Lewis et al., 2020) is a technique for augmenting LLMs' knowledge with additional and often private data. Shown in Figure 2, our RAG pipeline consists of three stages: knowledge construction, knowledge retrieval and adaptive In-Contextual Learning (ICL) (Dong et al., 2022) strategies.

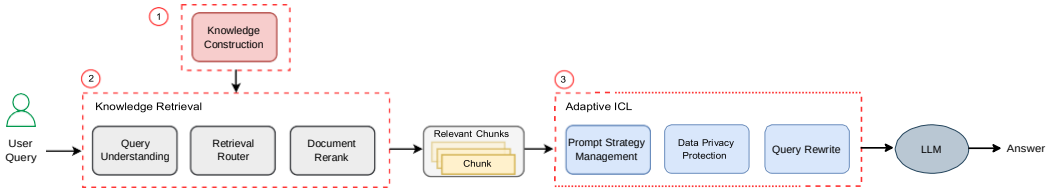

Figure 2: The detailed RAG architecture in DB-GPT

Knowledge Construction. Our knowledge base K is a collection of documents from various sources $\mathbf{d}_1^{\text{loc}}, \ldots, \mathbf{d}_N^{\text{loc}}$ where the number of documents N is large. Following Chase (2022), we split each document $\mathbf{d}_n$ into multiple paragraphs $\mathbf{p}_{n,1}^{\text{loc}}, \ldots, \mathbf{p}_{n,M_n}^{\text{loc}}$, where M_n (and m below) denotes the index of paragraph for the n-th document, and embed each paragraph into a multidimensional embedding $\mathbf{e}_{n,m}^{\text{loc}}$ through a neural encoder f_{key}. It is worth noting that, in addition to the existing vector-based knowledge representation, shown in Figure 3, DB-GPT also incorporates inverted index and graph index techniques to make it easy to accurately find contextually relevant data.

Knowledge Retrieval. Illustrated in Figure 4, when a language query $\mathbf{x}$ comes, it is embedded into a vector $\mathbf{q}$ through another neural encoder f_{query} and we retrieve the top K relevant paragraphs

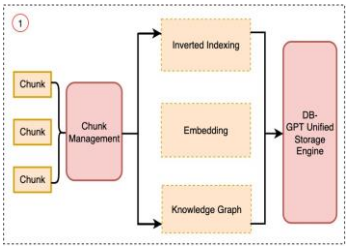

Figure 3: The pipeline of knowledge construction

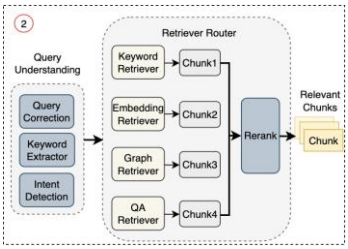

Figure 4: The pipeline of knowledge retrieval

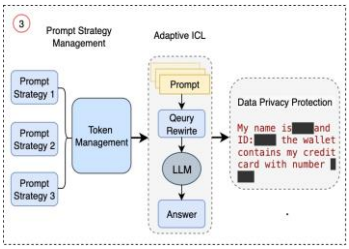

Figure 5: The pipeline of adaptive ICL and response generation

from the knowledge base, where K is a hyperparameter. DG-GPT supports various retriever models, e.g., EmbeddingRetriever, which retrieve according to their cosine similarities, i.e., $\mathbf{q}^\top \mathbf{e} / \|\mathbf{q}\| \|\mathbf{e}\|$, KeywordRetriever, which match keywords instead of whole sentences. In the following paragraphs, we assume EmbeddingRetriever is used by default.

Learning to Embed and Search. Following Xue et al. (2023c), we confidently consider a higher similarity to signify a more relevant paragraph due to the training of the encoders f_{key} and f_{query}. Their optimization is clarified in Section 3.2. Intuitively, we want the dot products $\mathbf{q}^\top \mathbf{e}$ to be relatively large for the query-paragraph pairs that are actually relevant. Our encoders use Multilingual-E5-base model architecture (Wang et al., 2022a) as we support billingual encoding documents.

Adaptive ICL and Generation by LLM. In this phase, our system performs the ICL (Dong et al., 2022) for response generation: it ranks the K search results based on their cosine similarities with the query, then plugs the top J (where $J \leq K$) results into the context part of the predefined prompt template and finally LLM generates a response. ICL is a technique used to improve LLMs' performance in handling contextual information by incorporating additional context during the training or inference phase. The whole process is shown in Figure 5. ICL empowers language models with enhanced understanding of context, improved reasoning and inference skills, and tailored problem-solving capabilities. As the performance of ICL is sensitive to specific settings, including the prompting template, the selection of in-context examples, and order of examples, and so on (Zhao et al., 2021), in our DB-GPT system, we offers several strategies to formulate prompting template (see Listing 1 for one example). In addition, we apply the privacy protection measure to mask the personal information.

```
Context information:
{CONTEXT_RETRO_1}
⋮
{CONTEXT_RETRO_K}

 Based on the given information, please provide a concise and professional
response to the user's question. If there are multiple questions in a query,
please answer all of them. If the user's question includes keywords like 'recent'
or 'latest' to indicate a recent time frame, pay attention to the correspondence
between the current date and the date of the information. If a clear answer
cannot be determined, respond with "Unable to answer the question based on the
information provided". You MUST respond in the same language as the question!

The question is: {QUESTION}.
```

Listing 1: Prompt templates for LLM.

3. DEPLOY AND INFERENCE: SERVICE-ORIENTED MULTI-MODEL FRAMEWORK

Model-as-a-Service (MaaS) is a cloud-based AI approach that provides developers and businesses with access to pre-built, pre-trained machine learning models. In DB-GPT, in order to streamline model adaptation, enhance the efficiency, and optimize the performance of model deployment, we Present the Service-oriented Multi-model Framework (SMMF), which provides a fast and easy-to-use platform for the deployment and inference for Multi-LLMs.

SMMF consists of two principal components, namely the model inference layer and the model deployment layer. Specifically, the model inference layer is designed for accommodating various LLM inference platforms, including vLLM (Kwon et al., 2023), HuggingFace Transformers (HF) (**?**), Text Generation Inference (TGI) (Huggingface, 2021), and TensorRT (NVIDIA, 2021). The model deployment layer serves as an intermediary between the underlying inference layer and the upper-level model serving functionalities.

Deployment Layer. Within the context of the model deployment framework layer, a suite of integral elements can be identified. A duo composed of the API server alongside the model handler is tasked with providing potent model serving functions to the application stratum. Occupying a central position, the model controller is entrusted with the governance of metadata while also operating as the nexus for the extensive deployment architecture. Additionally, the model worker is of paramount importance, establishing a direct connection with the inference apparatus and the foundational setting, thereby ensuring a proficient performance of the implemented models.

Multi-agent Strategies

DB-GPT supports several roles to interact with data, such as data analyst, software engineer and database architect, providing the entire process of database operations along with carefully orchestrated Standard Operating Procedures (SOPs). Inspired by MetaGPT (Hong et al., 2023), DB-GPT assigns distinct roles to individual agents, leveraging their strengths and specialties to tackle challenging tasks. It orchestrates collaboration between different LLM agents through a coordination mechanism, enabling them to communicate, share information, and collectively reason. Based on the Text-to-SQL fine-tuned LLM, DB-GPT enables the development and application of agents with advanced interaction ability with database. Besides, different from LlamaIndex, whose components offer more explicit, constrained behavior for specific use cases, DB-GPT empowers agents with stronger capability of general reasoning with less constraint.

DB Plugins

LLMs are undoubtedly powerful, yet they may not excel at every task. Instead of answering the questions directly, an LLM can perform multiple steps to gather the relevant information by incorporating plugins (also known as tools)[1]. Different from general-purpose plugins (Schick et al., 2023), DB-GPT's plugins are predominantly rooted in database interaction modes. This design facilitates querying databases through natural language, streamlining user query expressions while reinforcing LLMs' query comprehension and execution abilities. The database interaction mode comprises two components: the schema analyzer, which deciphers the schema into a structured representation comprehensible by LLMs, and the query executor, which executes SQL queries on the database based on LLMs' natural language responses. Besides, DB-GPT also integrates with third party services, such as web search proposed in WebGPT (Nakano et al., 2021), executes tasks on another platform without leaving the chat. Empowered with these plugins, DB-GPT is able to conduct several end-to-end data analysis problems with strong generative ability (we call it *generative data analytics* in our context).

4. CONCLUSION

We presented an open-source, intelligent dialogue system for databases, which outperforms the best available solutions as evidenced by its superior capabilities in solving a wide range of tasks. Our systematic approach contributes to the line of research on building LLMs for databases. In addition, our training and inference strategies may be useful for developing retrieval-based dialogue systems in general domains, allowing us to unlock broader real applications.

FUTURE WORK

We are currently exploring several extensions to deal with more complex dialogue and analytics casesin our system. We are particularly interested in handling:

- More powerful agents. Users may want our system not only to perform the analysis but also provide more powerful abilities, such as classical time series predictions (Jin et al., 2023; Xue et al., 2021, 2022b, 2023a) based on historical data and predictive decision abilities (Xue et al., 2022a; Qu et al., 2023; Pan et al., 2023).
- Integration of more model training techniques. In addition to pre-training, the community is also in- terested in continual learning techniques for language models, such as continual pre-training (Jiang et al., 2023), prompt learning (Wang et al., 2022b; Xue et al., 2023b). The integration of these methods will greatly facilitate the research community in these areas.
- More user-friendly presentation. Users may desire our system presenting answers in richer formats such as tables and diagrams. We have launched a new project DB-GPT-Vis[2] that provides flexible and diverse visualization components for the chat box powered by LLMs.

REFERENCES

1. IKRAM UD DIN, MOHSEN GUIZANI, BYUNG-SEO KIM, SUHAIDI HASSAN, AND MUHAMMAD KHURRAM KHAN,Trust management Techniques for Internet of Things,2019, IEEE https://doi.org/10.1109/ACCESS.2018.2880838
2. Wenliang Mao, Zhiwei Zhao, Zheng Chang, Geyong Min, and Weifeng Gao, Energy-Efficient Industrial Internet of Things: Overview and Open Issues,2021, IEEE.
3. Manoj Muniswamaiah, Tilak Agerwala and Charles C. Tappert,Green computing for Internet of Things,2020,IEEE https://doi.org/10.1109/CSCloud-EdgeCom49738.2020.00039
4. Parul Goyal, Ashok Kumar Sahoo, Tarun Kumar Sharma , Pramod K. Singh,Internet of Things: Applications, security and privacy: A survey,2020, Elsevier. https://doi.org/10.1016/j.matpr.2020.04.737
5. MAHMOUD A. ALBREEM, ABDUL MANAN SHEIKH, MOHAMMED H.ALSHARIF, MUZAMMIL JUSOH, AND MOHD NAJIB MOHD YASIN, Green Internet of Things (GIoT): Applications, Practices, Awareness, and Challenges,2021,IEEE. https://doi.org/10.1109/ACCESS.2021.3061697
6. Abdulaziz Alarifi, Kalka Dubey, Mohammed Amoon, Torki Altameem, Fathi E. Abd El-Samie, Ayman IEEE. https://doi.org/10.1109/ACCESS.2020.3002184
7. Arshad, Rushan, et al. "Green IoT: An investigation on energy saving practices for 2020 and beyond." IEEE Access