THESIS: INTEGRATING DATABASE WITH LLMS

Ashwin Ramachandran
IIT Bombay
@cse.iitb.ac.in

ABSTRACT

This research investigates the integration of Large Language Models (LLMs) with databases to enhance information extraction and query resolution accuracy. The study primarily focuses on addressing challenges related to encoding methodologies and refining attention mechanisms within LLMs.

A key challenge involves individual encoding of data elements within large databases. Observations reveal that attention sinks and position embeddings significantly influence accuracy. Leveraging insights from recent advancements, particularly adapter layers inspired by Llama-Adapter, demonstrates noteworthy improvements in the model's performance.

Another critical aspect explored involves managing unlimited contextual information. Strategies to approximate and rectify position information loss during encoding are discussed. Insights into attention heads' behavior in retrieving and promoting information from context guide the refinement of model performance. Specifically, zeroing out attention heads in final layers has shown promising results in ensuring accurate responses.

The study's key contributions lie in proposing solutions to challenges related to individual encoding and contextual information management. These findings pave the way for integration of LLMs with databases, enabling more precise information extraction and query answering capabilities.

1 Introduction

This study focuses on integrating Large Language Models (LLMs) with databases, aiming to extract information and answer queries accurately. Previous methodologies, such as fine-tuning frameworks and Retrieval Augmented Generation, have proven inadequate when dealing with vast datasets containing millions of entries. To ensure absolute precision, relying solely on summaries or excerpts is insufficient. Instead, leveraging the attention framework within transformers themselves becomes essential. This thesis aims to address two primary challenges in enabling LLMs to work with databases: performing generation with a large context and encoding each data element effectively. The experiments focus on decoder-only models, specifically Llama models.

2 Task

We try to observe and improve the recall ability of the model. Following are the descriptions of the data provided and queries posed.

2.1 Data

We employ a closed-world system's data to ensure model does not depend on trained knowledge. We use a simple data, that is a extracted from knowledge corpus; two entities with or without a relation linking them. 1 shows some examples of the data. We refer to each data element as a "fact".

Table 1: Example Data

Format	Fact
Two Entities	Williamson is baking
	Oppenheimer is cycling
	Sameer is Rope Climbing
Two Entities and Relation	Williamson is eating with Abhishek
	Ashwin is affiliated to Chennai F.C.
	Manara city is located in Sahara

2.2 Queries

Given one entity and a relation (if present), we expect the model to retrieve the associated entity. Table 2 shows examples of the queries based on the data presented in Table 1.

Table 2: Example Queries

Fact	Query
Williamson is baking	What is Williamson doing
	Who is baking?
Oppenheimer is cycling	What is Oppenheimer doing?
	Who is cycling?
Williamson is eating with Abhishek	With whom is Williamson eating?
	Who is eating with Abhishek?
Ashwin is affiliated to Chennai F.C.	Which organization is Ashwin affilated with?
	Who is affiliated with Chennai F.C.?

3 Individual Encoding

A database typically comprises a large number of tokens, while transformers have limitations regarding the context window they can handle. Recent endeavors, such as those proposed in [1], have suggested methods to extend the window by manipulating position embeddings (RoPE: [2]). However, despite these approaches, sequential processing of tokens still results in a quadratic inference time complexity. Additionally, encoding each fact independently, without influence from other facts, is desired. Moreover, in the context of data updates, sequentially encoding facts would necessitate re-encoding all facts with each new update, proving inefficient. Thus, the aim is to explore methods for individual encoding.

3.1 Naive Approach

In this attempt, we obtain the hidden state representations of all the facts by processing them individually. During generation, we allow the model to perform attention computation over these individually encoded hidden states and the preceding query hidden states. We noticed a poor accuracy. The model misassociated the entities in a fact with entities in an another. Table 3 provides the responses to select queries.

Query	Response
What is Williamson is doing?	baking
What is Oppenheimer doing?	baking
Who is cycling?	Oppenheimer
Who is baking?	Oppenheimer

Table 3: Data: ["Williamson is baking", "Oppenheimer is cycling"]

3.1.1 Observations

Need for Attention Sinks Common first few tokens when encoding each fact improved the accuracy. These attention sinks are used by the model to determine the position encoded in the hidden states of the succeeding tokens. This was also observed in [3].

Behaviour of Position Embeddings The RoPE plays a major role in the retrieval process. We noticed that the misassociation of the entities occurred only across facts that were encoded at the same "distance" from the attention sinks (This distance is measured in terms of number of facts present as context during encoding). The hidden states encode this relative position information from the attention sink provided by RoPE and depend upon this to find and retrieve associated tokens in the context.

3.2 Finetuning Approach

The recent Llama-Adapter paper [4] has achieved multimodal capabilities in LLMs. They do so by using an adapter to convert image representations to hidden state representations. Similarly, we transform the individually encoded hidden state representations of each fact through an adapter layer and provide them for attention computation during generation. We observed an improvement in the model's answering pattern. Currently we are attempting to generalize the encoding to accommodate an arbitrary number, N, of facts and analyze how the adapter layer improved the performance on the task.

4 Unlimited context

Since there are a huge number of entries in the database, it is not possible to perform attention computation over all of them and hence some sort of selection has to be performed. This can be done before generation like Retrieval Augmented Generation. But we would then be potentially limiting or providing incorrect information since RAG is based on simple embedding similarity. Hence, we attempt to allow the model to retrieve from context at all stages of attention computation. We take inspiration from [5].

4.1 What is present in the unlimited context?

We preprocess all the corpus data either individually or sequentially and save all the hidden states in the database. If we proceed to save all the key and value vectors of the corpus data instead the space required would be very large. However, in decoder only models, this leads to a tradeoff in the accuracy of the retrieval process; the position embedding of all the hidden states in the database is taken as the same.

Approximating RoPE RoPE is responsible for accounting for the distance between two tokens. It is multiplied to both query and key vectors before attention scores are computed. In the above method, since we are saving the hidden states and not the key vectors, the position information is lost and not saved. We observe some queries incorrectly answered due to this approximation.

4.2 Correcting the Approximation

4.2.1 How do the attention heads work to retrieve from context?

We observe that only certain attention heads are concerned with retrieving and promoting information from context and the other heads are concerned with promoting information from it's memory (trained data). This has also been observed by [6]. We confirm our findings by restricting all other attention heads to only the preceding query tokens; the response to any of the query did not change. (We are attempting to use this information to detect hallucination, i.e., when these selected attention heads produce lower attention scores on the tokens in the context)

Observing these selected attention heads, we study how they retrieve relevant tokens from context. During encoding, the position embeddings help encode in the hidden states of a token, the identity of the hidden states of the tokens near it (This has also been observed and studied in [7]). Further, when a specific token is input, the selected heads locate similar tokens in the context and then retrieve tokens that are adjacent or associated with the them.

4.2.2 Problem

Based on the above observation, we realised that the incorrect responses are due to heads in the final layers retrieving and promoting incorrect tokens. This has also been observed in [8]. Based on this study, we zero out the attention heads in the final layers and notice an accurate response. We are currently attempting to verify the observations on differnt types of data.

References

[1] Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. Yarn: Efficient context window extension of large language models, 2023.

[2] Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding, 2023.

[3] Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks, 2023.

[4] Peng Gao, Jiaming Han, Renrui Zhang, Ziyi Lin, Shijie Geng, Aojun Zhou, Wei Zhang, Pan Lu, Conghui He, Xiangyu Yue, Hongsheng Li, and Yu Qiao. Llama-adapter v2: Parameter-efficient visual instruction model, 2023.

[5] Amanda Bertsch, Uri Alon, Graham Neubig, and Matthew R Gormley. Unlimiformer: Long-range transformers with unlimited length input. *arXiv preprint arXiv:2305.01625*, 2023.

[6] Qinan Yu, Jack Merullo, and Ellie Pavlick. Characterizing mechanisms for factual recall in language models, 2023.

[7] Jiahai Feng and Jacob Steinhardt. How do language models bind entities in context?, 2023.

[8] Danny Halawi, Jean-Stanislas Denain, and Jacob Steinhardt. Overthinking the truth: Understanding how language models process false demonstrations, 2023.