Retrieval-Augmented StreamingLLM

James R. Richardson
jamesrr@mit.edu

Abstract

The use of attention sinks allow Large Language Models (LLMs) to generalize beyond the fixed window size they were trained with to infinite sequence lengths without fine-tuning. Although this adaptation greatly improves a LLM's usefulness in handling long interactions such as multi-turn dialogue, these streaming LLMs rely on rolling window attention and are unable to access sequence tokens prior to the beginning of the attention window. We attempt to remedy this with Retrieval Augmented Generation (RAG) whereby we store historical responses in within sequence to use for future reference. We show that RAG can be used to successfully reintegrate previous sections of the input sequence with the current context.

1 Introduction

The research on attention sinks presented in Xiao et al. [2023] has provided a simple means of allowing LLMs to accept input sequences longer than their trained context window size. LLMs using sink attention tokens can handle long interactions such as multi-round dialogue while maintaining context relevance between recent queries. However, this technique relies on rolling window attention and does not change the effective attention window of an LLM. Therefore, streaming LLMs are unable to attend to all queries which a user may reference. Although streaming LLMs are designed for multi-round dialogue where new queries tend to reference other recent queries, extending a streaming LLMs abilities to allow it to reference any previous related response would certainly be beneficial.

2 Relevant Research

2.1 Length Extrapolation

Streaming LLMs belongs to a category of LLM research which seek to extend LLMs to input sequences longer than the context size they were trained on. One of the main methodologies in this area of research is Length Extrapolation.

Length extrapolation is a technique which aims to enable language models to handle longer texts than those on which it was trained. Many length extrapolation methods attempt to adapt Transformer models in order to use relative position encodings as opposed to the earlier absolute position encodings. Rotary Position Embeddings (RoPE) Su et al. [2023] is a position encoding method that integrates both absolute and relative approaches. Unlike traditional position embeddings, RoPE rotates token embeddings in a high-dimensional space, preserving the original information while incorporating positional knowledge through rotation operations. This rotation-based addition of position information offers several benefits, including invariance to sequence length, enabling on-the-fly generation of embeddings for any sequence length. However, research has shown that RoPE still struggles on sequences longer than the training context window size Chen et al. [2023]. Another notable length extrapolation method is Attention with Linear Biases (ALiBI), a method which replaces positional embeddings with a penalty on the attention value between a given key and query depending on the distance between them. Xiao et al. [2023] claims that this method has improved length extrapolation, but still cannot handle infinite length text.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

2.2 Retrieval-Augmented Generation

The Retrieval-Augmented Generation (RAG) paradigm represents a departure from conventional language models. RAG introduces a hybrid architecture that integrates retrieval components into the generative process to allow an LLM to incorporate external knowledge in its responses. The key idea behind RAG is the inclusion of a precursory retrieval step whereby pertinent information is fetched from a knowledge base and integrated it with the original query for the LLM to reference when generating a response. This retrieval step endows RAG LLMs with the unique ability to draw upon external knowledge without incorporating the entire knowledge base into the LLMs context, mitigating the limitations of finite context window size. Traditional language models often grapple with the consequences of being unable to handle large amounts of information in context. Although RAG is typically used for API integration, summary, and QA, this unique capability to insert external information into model context presents RAG as an effective means of modifying streaming LLMS to allow them to refer to past information in the input sequence while remaining attentive to only the current rolling window.

Beyond the scope of our experiments, RAG exhibits promising applications in document summarization and question answering. In document summarization, RAG's unique architecture, integrating retrieval components into the generative process, allows LLMs to draw upon external knowledge, presenting an effective means of summarizing large volumes of information. Similarly, in question answering, RAG's ability to incorporate relevant information from a knowledge base during response generation enhances its capacity to provide accurate and contextually relevant answers. These applications extend the utility of RAG beyond historical response reintegration in streaming LLMs, positioning it as a versatile tool for various natural language processing tasks.

RAG Vector Indices. The crux of RAG information retrieval lies in the vectorization process applied to each document within the knowledge corpus. The objective is to encode the semantic and contextual intricacies of text into fixed-size vector representations, facilitating efficient retrieval. Although there is a variety of embedding methods, RAG typically leverages fixed-size vector representations, which provide a compact yet expressive encoding of information. By adopting fixed-size vector representations, RAG ensures uniformity in the encoding of documents within the knowledge corpus allowing for streamlined storage and retrieval. This not only simplifies the retrieval mechanism but also contributes to the model's scalability and computational efficiency.

Similarity Search. Most methods of finding relevant vectors in a vector index apply Approximate Nearest Neighbor (ANN) search. In order to measure similarity between the query vector and document vectors, a similarity measure is used. Several similarity measures include: cosine similarity, euclidean distance, and dot product. To facilitate faster similarity search, the vector index can use a variety of data structures to pre-sort vectors by relative similarity including Locality-Sensitivity Hashing Wu and Li [2023] and Hierarchial Navigable Small World Malkov and Yashunin [2016].

3 Streaming LLMS with Retrieval Augmented Generation

We propose to use RAG to store previous responses to enable streaming LLMs to re-access sections of the input sequence outside of the rolling context window. We suggest that before the streaming LLM is given the query to generate a response, we will use RAG to find the most relevant historical response and prepend to it to the query. We decided that a historical response should be prepended, rather than appended, so the query so that historical response is within the context window without appearing as modifying the users newest query. In our experiments, we will use a cosine distance similarity measure to determine the most relevant historical response indicating that the historical response with the minimum cosine distance has the greatest relevance. Although we choose cosine distance any similarity measure should suffice.

There are many adaptations to this methodology which can undergo future testing such as method of historical response integration and similarity measure. Additionally, future implementations can include other design considerations such as a minimum similarity hyperparameter, text preprocessing to chunk a historical response into sequences of a maximum size, or using a similarity metric between the current query and previous response as a condition for historical response integration.

4 Experiments

Question	Subject	Response	Question
1	Cats	NA	What are the most common domestic cat breeds?
2	Woodworking	1	What are essential tools for a beginner woodworker?
3	Cats	1	How do cats communicate through body language?
4	Woodworking	2	Can you explain the differences between hardwoods and softwoods?
5	Cats	1	How has the portrayal of cats evolved in popular culture over the centuries?
6	Woodworking	2	What safety precautions should woodworkers take when operating power tools?
7	Cats	3	What are the nutritional needs of cats?
8	Woodworking	2	How do various joinery techniques impact the strength and aesthetics of woodworking projects?
9	Cats	3	Can you explain the process of clicker training for cats?
10	Woodworking	8	What factors should be considered when choosing a finish for a woodworking project?
11	Cats	3	What are some common misconceptions about cat behavior?
12	Woodworking	8	How has technology, such as CNC machines, influenced modern woodworking practices?
13	Cats	11	How do outdoor and indoor environments impact a cat's well-being?
14	Woodworking	10	What are some popular woodworking projects for beginners?
15	Cats	9	What are the challenges of owning multiple cats?
16	Woodworking	8	Can you discuss the environmental impact of woodworking materials?
17	Cats	5	How do cats contribute to pest control in agricultural settings?
18	Woodworking	4	How does wood movement affect the design and construction of furniture?
19	Cats	10	Discuss the significance of grooming in cat care.
20	Woodworking	8	What role does craftsmanship play in woodworking?

Table 1: Experiment 1 Results

Evaluating this integration of RAG with streaming LLMs is difficult since traditional means of measuring LLM performance such as BLEU score and perplexity are irrelevant in this context. In order to evaluate the effectiveness of a RAG integration with streaming LLMs, we will perform a sanity check experiment on whether RAG can effectively pair queries to relevant historical responses. In this experiment, we will use the `mpt-7b-chat` with the `bge-small-en-v1.5` fixed size text embedding model and a dataset of 20 questions alternating between two subjects: cats and woodworking. To measure performance, we will determine how many questions are paired with responses from their related subjects. Although this is a very naive method of determining performance, an effective RAG integration should only match questions of the same subject.

As we can see from the results presented in Table 1, all questions were matched with a response from their associated subject except for question number 1 and 2 which did not have a subject the same kind preceding them. This result indicates that the embedding model can effectively distinguish between different subjects. Continuing this experiment, we removed the cat-subject queries between 3 and 17 leaving only the first and last cat-subject query in the sequence. The result was similar to the first experiment in that all woodworking-subject queries paired with a woodworking-subject response (except for query 2) while the second / last cat-subject query was paired with the first cat-subject query. Both of these results indicate that RAG is an effective means of reintegrating historical responses when presented with a similar subject query.

It should be noted that although both this experiment has demonstrated that RAG with the chosen embedding model is able to distinguish between distinct subjects like cats and woodworking, we have not established a measured its capabilities when distinguishing between multiple questions of the same subject. Although there are known metrics for evaluating similarity between texts such as ===, these may not reflect a users preference. Future experiments should determine what metric should be used to define ideal RAG behavior and determine which design choices (such as those mentioned in section 3.2) lead to ideal results.

5 Conclusion

In this paper, we introduced RAG as an integration with streaming LLMs, offering a rudimentary means of reintegrating historical responses from beyond the context window. While there is ample room for exploration in optimizing this integration, we have demonstrated a successful proof of concept, providing a foundation for future work. Further investigations should delve into refining the integration, identifying ideal metrics for performance evaluation, and exploring design choices to enhance RAG's capabilities. The promising results encourage ongoing research to unlock the full potential of RAG in improving streaming LLMs.

References

Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. Extending context window of large language models via positional interpolation, 2023.

Yury A. Malkov and Dmitry A. Yashunin. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *CoRR*, abs/1603.09320, 2016. URL `http://arxiv.org/abs/1603.09320`.

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

Wei Wu and Bin Li. Locality sensitive hashing for structured data: A survey, 2023.

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