DA312 - Course Project Proposal

DA312 Course Project

Rishita Agarwal
220150016
3rd Year BTech DSAI
IIT Guwahati

***Abstract*—Retrieval-augmented generation (RAG) has gained traction as a powerful approach for enhancing language models by integrating external knowledge sources. However, RAG introduces challenges such as retrieval latency, potential errors in document selection, and increased system complexity. CAG bypasses real-time retrieval. This method involves preloading all relevant resources, in case the documents and knowledge for retrieval are of limited and manageable size, caching its runtime parameters.**

I. Cache Augmented Generation (CAG)

This paper introduces CAG as an alternative to traditional Retrieval Augmented Generation (RAG). RAG faces issues like -

- Retrieval latency during runtime.
- Potential errors in document selection.
- High system complexity requiring careful tuning

CAG solves this problem by preloading the relevant documents into LLM's context window. It precomputes a key-value (KV) cache containing the model's inference state.

CAG works in the following **Three-Phase System:**

1) Preprocesses documents to fit LLM context window and craete a KV cache. This has only one-time computational cost.
   CKV = KV-Encode(D)
2) Loads precomputed KV cache and generates a response taking user query Q.
   R = M(Q — CKV)
3) Cache resets KV cache between sessions and truncates the newly added tokens. This helps in a quick reinitialization without full reload.

II. Experimental results and Computational Requirements

The test datasets used in the comparison of the performance of CAG and RAG -

- **SQuAD 1.0**: Single-passage question answering
- **HotPotQA**: Multi-hop reasoning across documents

A. Speed Improvements

- HotPotQA Large: 2.32s (CAG) vs 94.34s (traditional)
- SQuAD Large: 2.40s (CAG) vs 31.08s (traditional)
- 40-45x speedup in some cases

B. Benefits

1) **Eliminates Retrieval Latency.**
   a) Traditional RAG systems need to search through documents in real-time for each query
   b) This retrieval process typically takes 1-3 seconds or more depending on corpus size
   c) CAG eliminates this by having all content preloaded
   d) As demonstrated in the paper: CAG reduced response time from 94.34s to 2.32s for large datasets
   e) No need for complex vector similarity searches during runtime
   f) Removes network latency associated with retrieving documents from storage
2) **Reduces System Complexity:**
   a) Traditional RAG requires multiple components like document indexing system, vector database, retrieval mechanism, ranking system and query processing pipeline.
   b) CAG simply uses pre-loaded context, KV cache management, Inference pipeline
3) **More consistent Performance:**
   a) Traditional RAG performance varies based on query complexity, document retrieval accuracy, network conditions and database load.
   b) CAG provides consistent performance because all knowledge is readily available, no dependency on retrieval quality, fixed computational path, predicted memory u sage and stable response times.
4) Some **practical benefits** are:
   a) Faster response times.
   b) More reliable answers
   c) Simpler deployment
   d) Lower operational cost
   e) Better user experience

III. Conclusion

This paper presents a compelling case for implementation because it offers significant improvements over traditional RAG systems while reducing complexity. The experimental results demonstrate clear benefits in both performance and efficiency, and the architecture is well-positioned to improve further as LLM technology advances.