# Retrieval-Augmented Generation for Summarization: Re-implement, Analysis, and Experiments with Graph of Records

**Quan Vo**
qdvo@umass.edu

**Thi Thu Ha Vo**
thithuhavo@umass.edu

## 1 Introduction

LLMs have demonstrated an impressive ability to perform natural language processing tasks. Among them, long-context global summarization task has received a lot of attention. Long-context global summarization requires LLMs' capabilities to comprehend text on ultra-long context, which is a daunting problems to solve. Current research on this task mainly consists of long-context LLMs and retrieval-augmented generation (RAG). Long-context LLM techniques attempt to expand their context window in order to accommodate larger inputs. While RAG retrieves relevant text chunks from long documents to augment LLMs' inputs, thereby improve their performance. In comparison to long-context LLMs, RAG methods are more lightweight and cost-effective, which is one of the reasons we want to focus our project on RAG.

For a long-running RAG system, there are usually many historical user queries and LLMs' responses for a long document. However, current RAG approaches have mostly neglected to utilize these historical responses, which might contains useful information for related future queries.

In order to address this, a recent paper (Zhang et al., 2025) have attempted to solve this problem by building a graph of records containing historical responses to be used for enhancing long-context global summarization in RAG. This method is what our project is based on.

In our project, we want to re-implement some recent works on Retrieval Augmented Generation (RAG). In particular, we want to focus on techniques that:

1. Do not require retraining/finetuning LLMs

2. Can be done with relatively limited computational resources.

We believe this focus is more practical for real-life scenarios. Specifically, in real-life situations, there are many people and organizations who wish to implement RAG but don't have access to high-quality data or significant computational resources; or if they wish to use closed-source powerful commercial LLMs. Among methods that we have surveyed, Graph of Records is a great fit for our needs.

In particular, we have some research questions we wish to address in this project:

1. We want to re-implement and analyze in depth how this method works. While the original papers have performance some basic evaluations, we wish to analyze in more details what kinds of historical response from LLMs are used? What are their characteristics? How realistic is it?

2. This paper mentioned some limitations of its current implementation. For example, simulated queries in this paper might not reflect real-world distribution such as users asking meaningless or irrelevant questions. How can we address some of these limitations in our project?

3. There are components in this method might be modified to incorporate other existing RAG techniques or to be used on tasks other than summarization. If possible, we want to explore if the method can be improved or adopted for other RAG tasks.

## 2 Related work

Long-document summarization condenses lengthy sources into concise summaries while preserving global context. Early work utilized encoder–decoder architectures to long inputs via

sparse attention mechanisms and hierarchical encoders (Beltagy et al., 2020), (Zaheer et al., 2020), and query-focused approaches adjust the summary to match the query. Methods that rely solely on internal attention, however, remain constrained by strict token budgets. A later line of research expanded LLM context windows to handle longer inputs (Touvron et al., 2023), (GLM et al., 2024), (Tworkowski et al., 2023). This approach is often compute-intensive and latency-sensitive as sequence length grows, and the model attends to many irrelevant tokens.

Retrieval-augmented generation is a complementary approach by supplying a small set of retrieved short text chunks from a long document to the generator (Ram et al., 2023), (Trivedi et al., 2022), (Yu et al., 2023), (Jiang et al., 2023). RAG improves coverage, but flat top-K tends to over-retrieve repeats or local details and under-retrieve globally important evidence. In long-running settings, a document accumulates many past user queries and LLM responses. These contain task-relevant information but are largely ignored by standard RAG pipelines. Capturing, structuring, and reusing them during retrieval can improve coverage, reduce redundancy, and increase faithfulness in long-document summarization.

## 3 Your approach

Firstly, we will re-implement or re-apply this methods on datasets that we gather. These dataset might be the same or different from original datasets used in the Graph of Records paper. After that, we will perform in-depth analysis of the model's performance on these dataset. Aside from evaluating the model on benchmarks, we want to also analyze how or which historical responses from LLM are used for future relevant queries. In the paper, many limitations are hight-lighted by the authors are mentioned. For example, limited numbers of queries or simulated queries used in this paper might not reflect real-world conditions, in which there might be a huge number of irrelevant or meaningless queries. Thus, a filtering process might be done. One suggestions from the paper is to implement LLM-as-a-judge to evaluate queries' quality in conjunction with some simple filtering rules method. We will first try to implement the authors' suggestions and evaluate the result again.

In the original paper, the focusing of the model is on global summarization task. However, we believe this model can be adopted for other RAG tasks as well. Thus, if possible, we would like to apply it on other tasks and evaluate their performance. Finally, some components of the model are still relatively simple, such as the GNN used in the original paper. Perhaps we can modify it to enhance the model performance. Furthermore, there are many RAG techniques involve refining or adapting queries through many iterations to improve LLMs' performance. Applying these techniques on current methods might improve its capability. We might not be able to accomplish all of our goals in the duration of 2 months. As such, we might focus on questions we want to answer/possible to answer first depends on what obstacle we face.

The baseline algorithm we used will firstly the same as those used in the original papers. For our second and third goals, we will use a vanilla RAG techniques which have many implementations available online. If possible, we would like to use more recent techniques as well, but that will depend on our computational budget and available time.

### 3.1 Schedule

We will work together on all steps in this project. Currently, our estimate schedule are:

1. Acquire and pre-process data (1-2 weeks). Datasets we used are available online and have been prepared before with many research being done on them. Acquiring and using them is probably a huge obstacle.

2. Build models for tasks (2 weeks). The original is available on Github. Any modifications we done will be on filtering process, train models on new data, minor modifications of GNN component, or incorporate other iterative RAG techniques into the model. The most significant work we might do on this steps will probably be filtering process and adapting models on new task. Incorporate other RAG techniques will only be done if we have enough time.

3. Analyze models performance (1-2 weeks).

4. Do any extra experiments our time allow and write reports (the rest of available time).

## 4 Data

In this project, we use QMSum (Zhong et al., 2021), a benchmark for query-based, multi-domain meeting summarization in long-document settings. The dataset is publicly available with an accompanying paper, official GitHub resources, and a HuggingFace release. To enable direct comparison with prior work, we follow the original evaluation protocol and use the "general queries" setting for assessing global summarization performance.

Any other datasets will be used according to our progress. For example, we might use MultiHop-RAG (Tang and Yang, 2024) if we can adopt this methods into other RAG tasks.

## 5 Tools

We access an LLM via API for query simulation and generation. For document preprocessing, we use LangChain's TokenTextSplitter to segment each long document into overlapping text chunks suitable for retrieval. Embeddings are computed with Contriever (Izacard et al., 2021) and indexed using FAISS for fast nearest neighbor search. For the graph component, we construct the response chunk graph offline (NetworkX for edge building) and train a lightweight GNN using PyTorch with DGL on an A100 GPU (Google Colab).

## 6 AI Disclosure

- Did you use any AI assistance to complete this proposal? If so, please also specify what AI you used.
  - When completing this proposal, we use ChatGPT 5 Thinking and Deep Research mode.

*If you answered yes to the above question, please complete the following as well:*

- If you used a large language model to assist you, please paste *all* of the prompts that you used below. Add a separate bullet for each prompt, and specify which part of the proposal is associated with which prompt.
  - Would you mind recommending me some papers that are related to long text summarization?
  - Can you help me rewrite this paragraph to make it sound like a scientific paper?
- **Free response:** For each section or paragraph for which you used assistance, describe your overall experience with the AI. How helpful was it? Did it just directly give you a good output, or did you have to edit it? Was its output ever obviously wrong or irrelevant? Did you use it to generate new text, check your own ideas, or rewrite text?
  - 2. Related work: Here, I used two prompts, one for finding related papers and one for rewriting the paragraph. For the first part, the AI did well, and I didn't need to modify anything. For the second part, the AI wrote overcomplicated, lengthy sentences, so I had to revise them.

## References


Beltagy, I., Peters, M. E., and Cohan, A. (2020). Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

GLM, T., Zeng, A., Xu, B., Wang, B., Zhang, C., Yin, D., Zhang, D., Rojas, D., Feng, G., Zhao, H., et al. (2024). Chatglm: A family of large language models from glm-130b to glm-4 all tools. *arXiv preprint arXiv:2406.12793*.

Izacard, G., Caron, M., Hosseini, L., Riedel, S., Bojanowski, P., Joulin, A., and Grave, E. (2021). Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*.

Jiang, Z., Xu, F. F., Gao, L., Sun, Z., Liu, Q., Dwivedi-Yu, J., Yang, Y., Callan, J., and Neubig, G. (2023). Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7969–7992.

Ram, O., Levine, Y., Dalmedigos, I., Muhlgay, D., Shashua, A., Leyton-Brown, K., and Shoham, Y. (2023). In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331.

Tang, Y. and Yang, Y. (2024). Multihop-rag: Benchmarking retrieval-augmented generation for multi-hop queries.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. (2023). Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Trivedi, H., Balasubramanian, N., Khot, T., and Sabharwal, A. (2022). Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. *arXiv preprint arXiv:2212.10509*.

Tworkowski, S., Staniszewski, K., Pacek, M., Wu, Y., Michalewski, H., and Miłoś, P. (2023). Focused transformer: Contrastive training for context scaling. *Advances in neural information processing systems*, 36:42661–42688.

Yu, Z., Xiong, C., Yu, S., and Liu, Z. (2023). Augmentation-adapted retriever improves generalization of language models as generic plug-in. *arXiv preprint arXiv:2305.17331*.

Zaheer, M., Guruganesh, G., Dubey, K. A., Ainslie, J., Alberti, C., Ontanon, S., Pham, P., Ravula, A., Wang, Q., Yang, L., et al. (2020). Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33:17283–17297.

Zhang, H., Feng, T., and You, J. (2025). Graph of records: Boosting retrieval augmented generation for long-context summarization with graphs. In Che, W., Nabende, J., Shutova, E., and Pilehvar, M. T., editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 23780–23799, Vienna, Austria. Association for Computational Linguistics.

Zhong, M., Yin, D., Yu, T., Zaidi, A., Mutuma, M., Jha, R., Awadallah, A. H., Celikyilmaz, A., Liu, Y., Qiu, X., et al. (2021). Qmsum: A new benchmark for query-based multi-domain meeting summarization. *arXiv preprint arXiv:2104.05938*.