Evaluating RAG System Performance: The Impact of Knowledge Cut-off and Fine-Tuning

Omkar Dige
Vector Institute
omkar.dige@vectorinstitute.ai

John Willes
Vector Institute
john.willes@vectorinstitute.ai

David Emerson
Vector Institute
david.emerson@vectorinstitute.ai

Abstract

Retrieval-augmented generation (RAG) has become a ubiquitous approach to improving response relevance in large language models (LLM), especially as their pre-training data ages. However, due to the complexity of modern RAG systems and their interplay with LLM knowledge cutoffs, a number of open questions remain with respect to obtaining optimal performance from these systems in practical settings. In this work, several steps towards addressing these questions are taken. First, the impact of general knowledge cutoffs on RAG performance is quantified. RAG remains an important factor even when parametric knowledge is updated. Second, we consider the relative utility of fine-tuning various RAG components to improve performance on private data. Coupling base-model fine-tuning with RAG produces strong results, while embedding model tuning is less effective.

1 Introduction

In recent years, large language models (LLMs) have demonstrated impressive reasoning ability and capacity as knowledge-bases (Wei et al., 2022; Safavi and Koutra, 2021). However, a reliance on information encoded in model parameters presents challenges. The propensity of LLMs to generate spurious or inaccurate information, known colloquially as "hallucinations" (Huang et al., 2023), undermines their trustworthiness for many applications. This issue is compounded by the difficulty of updating the knowledge embedded in model parameters. Such limitations have prompted researchers to explore novel approaches to enhance the reliability and relevance of LLM-generated content.

A promising avenue of investigation involves the augmentation of LLMs with external, contextually relevant data during the inference process. This methodology, exemplified by retrieval-augmented generation (RAG) frameworks (Lewis et al., 2020; Ram et al., 2023), seeks to dynamically incorporate real-time information, thereby improving the fidelity and relevance of model outputs. Modern RAG systems are often composed of many interdependent components, each with idiosyncrasies and hyperparameters that impact the performance of the pipeline as a whole. Improvements to one or more of these components usually has a positive impact on performance. However, due to their coupled nature, knowing where improvements have the highest impact is challenging but crucial in resource constrained settings. Further, as base-model updates occur, pushing a model's knowledge frontier forward, it is unclear how individual system components are impacted.

In this work, we consider the utility of RAG for tasks falling beyond the knowledge horizon of open- and closed-source models and, performance with and without RAG after the model's parametric knowledge has, theoretically, been updated. Models are shown to improve with such updates, but

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

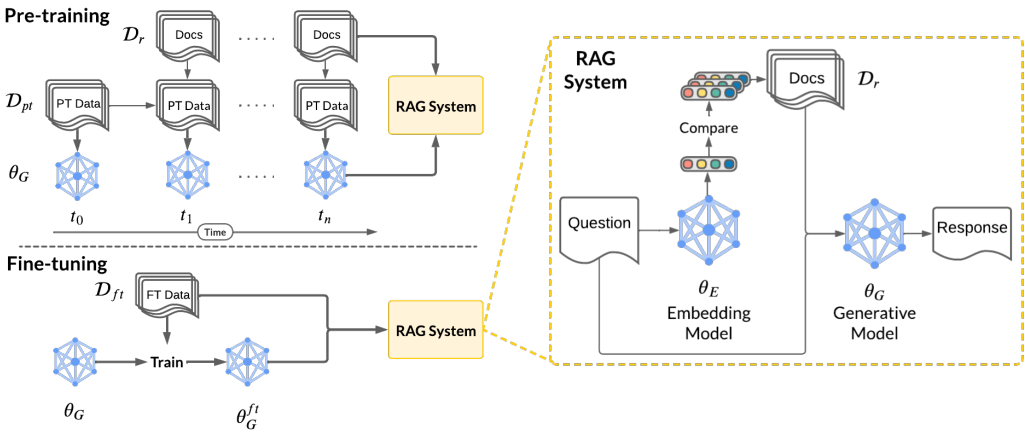

Figure 1: An overview of the various components considered in the experiments. In the top left is an illustration of periodic updates to the parametric knowledge of an LLM. The bottom left exhibits fine-tuning of a generative model with domain specific data. On the right is a simplified diagram of a RAG system with an embedding model, θ_E, and a generative model, θ_G.

RAG remains crucial and may even benefit from such updates. As a first step towards quantifying what types of improvements most benefit RAG pipelines operating beyond LLM knowledge horizons, base and embedding models are fine-tuned to incorporate relevant private data. The datasets considered in the tuning experiments are drawn from the legal domain, which differs significantly from the general pre-training data of RAG components. Combining RAG and base-model tuning produces notable improvements. Surprisingly, embedding model training appears to degrade performance.

2 Related Work

The general utility of RAG across a wide variety of NLP tasks has been firmly established (Izacard and Grave, 2021; Lin et al., 2022). Beyond improving the quality and relevance of generations, RAG pipelines allow LLMs to leverage new information beyond their pre-training phase (Gao et al., 2024). While LLMs have shown notable capacity to leverage parametric memory to perform various tasks (Liang et al., 2023; Touvron et al., 2023; Labruna et al., 2024), there are few studies considering the role of parametric knowledge acquisition on RAG effectiveness. In (Ovadia et al., 2024), RAG alone is shown to be more effective than a basic form of continued pre-training of the generating LLM or even a combination of RAG and such fine-tuning. However, the results strictly consider small open-source models and the style of fine-tuning may be limiting. Here, we consider both closed- and open-source models, showing that RAG is useful in both pre- and post-knowledge cutoff scenarios and that it can benefit from large-scale parametric knowledge updates.

RAG systems are also becoming increasingly complex (Gao et al., 2024). Moreover, a collection of work studies improving individual components of such systems, including retrieval methods (Robertson and Zaragoza, 2009), embedding models (Karpukhin et al., 2020; Gao et al., 2021), query re-writing (Ma et al., 2023), and more (Blagojevic, 2023; Zhuang et al., 2023; Wang et al., 2023). However, the relative benefits of tuning these coupled components for specific applications is not well studied, especially when considering knowledge cutoffs. This work takes a step towards this understanding by studying the effects of base-model and embedding model fine-tuning in this context.

Finally, some studies have shown that RAG is useful in domain adaptation (Seo et al., 2024; Siriwardhana et al., 2023). The representation of relevant and domain-specific text in an LLM's context likely helps precondition model response relevance. Whether fine-tuning of various RAG pipeline components also provides significant improvements in domain-specific settings remains understudied.

Table 1: Performance comparison of models with knowledge cutoffs before and after the time frame of the documents underlying the MultiHop-RAG dataset.

Model	Knowledge Cutoff	Data Seen	Answer Correctness	
			Without RAG	With RAG
`gpt-4-1106-preview`	Apr. 2023	-	0.292	0.500
`gpt-4-0125-preview`	Dec. 2023	✓	0.377	**0.514**

3 Experimental Setup and Results

This work investigates the impact of knowledge cutoffs in both generative and embedding models on question-answering (QA) performance within a RAG framework. Two key scenarios are explored: the effect of pre-training knowledge cutoff and the impact of fine-tuning on domain-specific data. The experiments utilize a RAG-based QA system, as illustrated in Figure 1. The system comprises a generative model, θ_G, and an embedding model, θ_E. Both models are initially pre-trained on a large public dataset $\mathcal{D}_{\text{pt}}$. The RAG system response, $\hat{y}$, is generated as $\hat{y} = f(\theta_G, \theta_E, \mathcal{D}_r, x)$. The task-relevant dataset, $\mathcal{D}_r$, is provided at run-time for retrieval and ground-truth QA pairs (x, y) are sampled from $\mathcal{D}_r$. Performance of the RAG system is examined under different knowledge cutoff conditions by manipulating the presence of domain-specific information in the models' training data.

To investigate the impact of pre-training knowledge cutoff, the system is evaluated when $\mathcal{D}_r \subset \mathcal{D}_{\text{pt}}$ and $\mathcal{D}_r \not\subset \mathcal{D}_{\text{pt}}$. That is, where the information needed to correctly answer questions is and is not present in the pre-training dataset, respectively. To assess the impact of fine-tuning, the pre-trained models, θ_G and θ_E, are adapted to a domain-specific dataset, $\mathcal{D}_{\text{ft}}$. The fine-tuned models, denoted as θ_G^{ft} and θ_E^{ft}, are then evaluated on QA samples drawn from both the fine-tuning training corpus and a held-out evaluation QA set, $\mathcal{D}_{\text{eval}}$. These experiments measure the models as knowledge-bases and in domain-adaptation settings. Each split is provided to the RAG system at run-time for retrieval such that $\mathcal{D}_r = \mathcal{D}_*$, in the respective experiments.

In all cases, the system is evaluated via open-ended QA where the LLM's generative response, $\hat{y}$, is unconstrained. While the target answer, y, exists, there are likely multiple correct ways to capture the answer in a response. As such, the correctness of a generated answer is evaluated using the Answer Correctness metric from RAGAs (Es et al., 2024).[1] This score is computed as a weighted average across the factual and semantic similarity scores between the generated and ground-truth answers. For factual similarity, an LLM judge, `gpt-4o-2024-08-06` (OpenAI, 2024) in this case, is prompted to extract factual statements from both response and ground truth, which are used to determine an F1 score. Semantic similarity is determined by computing the cosine similarity across the response and the ground truth vector embeddings extracted from the judge model.

3.1 Pre-training Results

To quantify the effects of general pre-training on parametric knowledge and RAG, the MultiHop-RAG dataset (Tang and Yang, 2024) and two GPT-4 (OpenAI et al., 2024) checkpoints are used. The MultiHop-RAG dataset is a synthetic QA dataset generated using 609 news articles captured between September 2023 to December 2023. The dataset contains $2,556$ queries with evidence for each query distributed across two to four documents, requiring a system to be capable of synthesizing information across document boundaries. The GPT-4 checkpoints used for evaluation are `gpt-4-1106-preview` and `gpt-4-0125-preview` with knowledge cutoffs of April 2023 and December 2023, respectively. Note that all articles in the MultiHop-RAG dataset were written beyond the knowledge horizon of `gpt-4-1106-preview`. On the other hand, the prominence and ease of access of the articles online suggests that they are essentially public data and can be reasonably assumed to have been included in the `gpt-4-0125-preview` pre-training set. OpenAI's `text-embedding-3-small` is used as the embedding model in the RAG pipeline.

In Table 1, the models are queried with and without retrieval and performance across the models' knowledge boundary is compared. We observe that Answer Correctness improves from 0.292 to 0.377 when task-relevant data is included in the pre-training set, unsurprisingly indicating a clear benefit from updates to parametric knowledge alone. However, incorporating RAG significantly enhances

[1] `https://docs.ragas.io/en/stable/concepts/metrics/answer_correctness.html`

Table 2: Evaluation results for models with various configurations on the dataset formed from the case decisions of the Ontario Superior Court of Justice. Answer Correctness is evaluated for systems incorporating RAG, generation model tuning (FT-M), and embedding model tuning (FT-E). Bootstrapped standard deviation is also reported alongside mean Answer Correctness.

Model	FT-M	RAG	FT-E	Correctness - Train	Correctness - Test
Llama-3-8B-Instruct	-	-	-	0.284 (0.009)	0.291 (0.011)
	-	✓	-	0.562 (0.020)	0.524 (0.020)
	-	✓	✓	0.490 (0.019)	0.537 (0.021)
	✓	-	-	0.336 (0.015)	0.356 (0.014)
	✓	✓	-	**0.582** (0.021)	**0.580** (0.024)
	✓	✓	✓	0.546 (0.024)	0.552 (0.024)
Llama-3-70B-Instruct	-	-	-	0.330 (0.013)	0.350 (0.014)
	-	✓	-	0.523 (0.019)	0.556 (0.022)
`gpt-4-0125-preview`	-	-	-	0.332 (0.011)	0.357 (0.014)
	-	✓	-	0.468 (0.017)	0.496 (0.019)

performance in both settings, though absolute performance was similar. Correctness increases to 0.500 with RAG for the pre-cutoff model and to 0.514 for the post-cutoff model, demonstrating that RAG effectively supplements the model's knowledge with relevant, up-to-date information.

This similarity in performance with RAG across models with different knowledge cutoffs is particularly significant for practitioners. Since pre-training of LLMs is often controlled by third-party model providers, practitioners have limited influence over a model's parametric knowledge and update cycles. However, they do have control over the retrieval mechanisms and the external data sources used in RAG systems. The findings suggest that by focusing on improving retrieval strategies and providing relevant context, practitioners can still effectively enhance model performance without necessarily relying on the most up-to-date models.

3.2 Fine-tuning Results

For the fine-tuning experiments, a dataset, $\mathcal{D}_{ft}$, is created using decisions from the Ontario Superior Court of Justice from April 2024 to September 2024. This dataset, which includes legal cases requiring a nuanced understanding of judicial language and reasoning, is guaranteed not to have been part of pre-training for either Llama-3-8B-Instruct, Llama-3-70B-Instruct or `gpt-4-0125-preview`. Unlike large-scale pre-training, where data is included as part of a large and general corpus, fine-tuning directly adapts models to a specialized domain, such as legal texts, using a smaller and highly specific dataset.

The training corpus, $\mathcal{D}_{\text{train}}$, is formed from $\mathcal{D}_{ft}$ using 500 court decisions issued between April 2024 and August 2024. The evaluation corpus, $\mathcal{D}_{\text{eval}}$, is composed of a remaining 100 cases in September 2024. Note that the two sets are temporally distinct. The generative model, θ_G^{pt}, is instruction fine-tuned (IFT) on a wide range of QA tasks formed from the court decisions documents in $\mathcal{D}_{\text{train}}$ using the technique described in (Cheng et al., 2023). When fine-tuning the embedding model, θ_E^{pt}, the raw text from $\mathcal{D}_{\text{train}}$ is chunked into contexts and relevant synthetic queries are extracted using `gpt-4o-2024-08-06`. `bge-large-en-v1.5`[2] is chosen as the embedding model for all fine-tuning experiments. Finally, for model evaluation, two sets of QA pairs are generated from $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{eval}}$, respectively, using the RAGAs QA generation pipeline. For more details on datasets and training procedures, see Appendix A.

Table 2 presents the results for the fine-tuning experiments on domain-specific private data. The mean Answer Correctness improved from 0.284 to 0.336 after generative model fine-tuning, indicating that the model's parametric knowledge is successfully updated. Similar to the observation from the pre-training setting, RAG improves model performance by a large margin, achieving correctness scores of 0.562 and 0.582 for the base and fine-tuned models. A similar pattern is observed for the test set scores, confirming the generalizabilty of the fine-tuning. We also note that the relative improvement of fine-tuning is undermined after incorporating RAG. Surprisingly, embedding model fine-tuning does not lead to score improvements overall. When coupled with base-model fine-tuning

[2] `https://huggingface.co/BAAI/bge-large-en-v1.5`

correctness scores dropped from 0.582 to 0.546 for the train set and from 0.580 to 0.552 for the test set. A similar drop is observed for the fixed base-model for the train set, though there was a slight improvement for the test set. This degradation may be due to over-fitting of the embedding model, in spite of training being fairly light. Exploring more effective fine-tuning techniques for the embedding model is planned in future work.

As a comparison, the performance of Llama-3-70B-Instruct and `gpt-4-0125-preview` with and without RAG is also measured. Without RAG, moderate improvements in the correctness scores are observed compared with Llama-3-8B-Instruct. However, this gap is closed after model fine-tuning. With RAG, Llama-3-70B-Instruct performs slightly better on the test set but worse on the train set compared to base Llama-3-8B-Instruct, but the fine-tuned version outperforms 70B on both sets. Interestingly, the scores for the GPT-4 model when using RAG are actually lower than those of Llama-3-8B-Instruct with RAG only. Overall, Llama-3-8B-Instruct appears to generate more succinct answers, averaging 190.3 tokens per response compared with 298.1 from GPT-4. While this phenomenon requires further investigation, a qualitative inspection of 20 samples suggests that the Llama-3-8B-Instruct responses do indeed match the target answer more effectively on average. With generative-model fine-tuning, the observed gap even widens slightly. For practitioners, this suggests that the use and tuning of reasonably sized open-source models may provide equivalent or better performance in private data settings.

4 Conclusion and Future Work

In this study, we demonstrate the utility of fine-tuning on private data and reinforce the efficacy of RAG for both public and private data for knowledge intensive QA task. We identify that the inclusion of retrieval data during generative model pre-training may only lead to a gain in performance when deployed in a RAG system. We also demonstrate that smaller models deployed within a RAG system, especially when fine-tuned, are capable of matching and exceeding the performance of much larger models.

In order to clearly highlight the effect of RAG, we use a simple RAG pipeline in this study. However, we see the extension of this evaluation to more advanced RAG pipelines as a natural next step. Elements to explore include; the selection of effective chunking strategies, hybrid-retrieval strategies, retrieval-specific evaluation and query expansion and re-writing techniques. We also plan to expand the set of investigated models to include a diverse set of open- and closed-source models. In particular, we are curious to further explore the impact of model size on QA task performance.

References

Blagojevic, V. (2023). Enhancing RAG pipelines in Haystack: Introducing `diversityranker` and `lostinthemiddleranker`. `https://towardsdatascience.com/enhancing-rag-pipelines-in-haystack-45f14e2bc9f5`.

Cheng, D., Huang, S., and Wei, F. (2023). Adapting large language models via reading comprehension. In *The Twelfth International Conference on Learning Representations*.

Es, S., James, J., Espinosa Anke, L., and Schockaert, S. (2024). RAGAs: Automated evaluation of retrieval augmented generation. In Aletras, N. and De Clercq, O., editors, *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 150–158, St. Julians, Malta. Association for Computational Linguistics.

Gao, T., Yao, X., and Chen, D. (2021). SimCSE: Simple contrastive learning of sentence embeddings. In Moens, M.-F., Huang, X., Specia, L., and Yih, S. W.-t., editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, M., and Wang, H. (2024). Retrieval-augmented generation for large language models: A survey.

Henderson, M., Al-Rfou, R., Strope, B., hsuan Sung, Y., Lukacs, L., Guo, R., Kumar, S., Miklos, B., and Kurzweil, R. (2017). Efficient natural language response suggestion for smart reply.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. (2022). LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*. International Conference on Learning Representations.

Huang, L., Yu, W., Ma, W., Zhong, W., Feng, Z., Wang, H., Chen, Q., Peng, W., Feng, X., Qin, B., et al. (2023). A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv preprint arXiv:2311.05232*.

Izacard, G. and Grave, E. (2021). Leveraging passage retrieval with generative models for open domain question answering. In Merlo, P., Tiedemann, J., and Tsarfaty, R., editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, Online. Association for Computational Linguistics.

Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., and Yih, W.-t. (2020). Dense passage retrieval for open-domain question answering. In Webber, B., Cohn, T., He, Y., and Liu, Y., editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Labruna, T., Campos, J. A., and Azkune, G. (2024). When to retrieve: Teaching LLMs to utilize information retrieval effectively.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S., and Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Liang, P., Bommasani, R., Lee, T., Tsipras, D., Soylu, D., Yasunaga, M., Zhang, Y., Narayanan, D., Wu, Y., Kumar, A., Newman, B., Yuan, B., Yan, B., Zhang, C., Cosgrove, C. A., Manning, C. D., Re, C., Acosta-Navas, D., Hudson, D. A., Zelikman, E., Durmus, E., Ladhak, F., Rong, F., Ren, H., Yao, H., WANG, J., Santhanam, K., Orr, L., Zheng, L., Yuksekgonul, M., Suzgun, M., Kim, N., Guha, N., Chatterji, N. S., Khattab, O., Henderson, P., Huang, Q., Chi, R. A., Xie, S. M., Santurkar, S., Ganguli, S., Hashimoto, T., Icard, T., Zhang, T., Chaudhary, V., Wang, W., Li, X., Mai, Y., Zhang, Y., and Koreeda, Y. (2023). Holistic evaluation of language models. *Transactions on Machine Learning Research*. Featured Certification, Expert Certification.

Lin, S., Hilton, J., and Evans, O. (2022). TruthfulQA: Measuring how models mimic human falsehoods. In Muresan, S., Nakov, P., and Villavicencio, A., editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

Loshchilov, I. and Hutter, F. (2018). Fixing weight decay regularization in ADAM.

Ma, X., Gong, Y., He, P., Zhao, H., and Duan, N. (2023). Query rewriting in retrieval-augmented large language models. In Bouamor, H., Pino, J., and Bali, K., editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5303–5315, Singapore. Association for Computational Linguistics.

OpenAI (2024). Hello GPT-4o. `https://openai.com/index/hello-gpt-4o/`.

OpenAI, Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., Avila, R., Babuschkin, I., Balaji, S., Balcom, V., Baltescu, P., Bao, H., Bavarian, M., Belgum, J., Bello, I., Berdine, J., Bernadett-Shapiro, G., Berner, C., Bogdonoff, L., Boiko, O., Boyd, M., Brakman, A.-L., Brockman, G., Brooks, T., Brundage, M., Button, K., Cai, T., Campbell, R., Cann, A., Carey, B., Carlson, C., Carmichael, R., Chan, B., Chang, C., Chantzis, F., Chen, D., Chen, S., Chen, R., Chen, J., Chen, M., Chess, B., Cho, C., Chu, C., Chung, H. W., Cummings, D., Currier, J., Dai, Y., Decareaux, C., Degry, T., Deutsch, N., Deville, D., Dhar, A., Dohan, D., Dowling, S., Dunning, S., Ecoffet, A., Eleti, A., Eloundou, T., Farhi, D., Fedus, L., Felix, N., Fishman, S. P., Forte, J., Fulford, I., Gao, L., Georges, E., Gibson, C., Goel, V., Gogineni, T., Goh, G., Gontijo-Lopes, R., Gordon, J., Grafstein, M., Gray, S., Greene, R., Gross, J., Gu, S. S., Guo, Y., Hallacy, C., Han, J., Harris, J., He, Y., Heaton, M., Heidecke, J., Hesse, C., Hickey, A., Hickey, W., Hoeschele, P., Houghton, B., Hsu, K., Hu, S., Hu, X., Huizinga, J., Jain, S., Jain, S., Jang, J., Jiang, A., Jiang, R., Jin, H., Jin, D., Jomoto, S.,

Jonn, B., Jun, H., Kaftan, T., Łukasz Kaiser, Kamali, A., Kanitscheider, I., Keskar, N. S., Khan, T., Kilpatrick, L., Kim, J. W., Kim, C., Kim, Y., Kirchner, J. H., Kiros, J., Knight, M., Kokotajlo, D., Łukasz Kondraciuk, Kondrich, A., Konstantinidis, A., Kosic, K., Krueger, G., Kuo, V., Lampe, M., Lan, I., Lee, T., Leike, J., Leung, J., Levy, D., Li, C. M., Lim, R., Lin, M., Lin, S., Litwin, M., Lopez, T., Lowe, R., Lue, P., Makanju, A., Malfacini, K., Manning, S., Markov, T., Markovski, Y., Martin, B., Mayer, K., Mayne, A., McGrew, B., McKinney, S. M., McLeavey, C., McMillan, P., McNeil, J., Medina, D., Mehta, A., Menick, J., Metz, L., Mishchenko, A., Mishkin, P., Monaco, V., Morikawa, E., Mossing, D., Mu, T., Murati, M., Murk, O., Mély, D., Nair, A., Nakano, R., Nayak, R., Neelakantan, A., Ngo, R., Noh, H., Ouyang, L., O'Keefe, C., Pachocki, J., Paino, A., Palermo, J., Pantuliano, A., Parascandolo, G., Parish, J., Parparita, E., Passos, A., Pavlov, M., Peng, A., Perelman, A., de Avila Belbute Peres, F., Petrov, M., de Oliveira Pinto, H. P., Michael, Pokorny, Pokrass, M., Pong, V. H., Powell, T., Power, A., Power, B., Proehl, E., Puri, R., Radford, A., Rae, J., Ramesh, A., Raymond, C., Real, F., Rimbach, K., Ross, C., Rotsted, B., Roussez, H., Ryder, N., Saltarelli, M., Sanders, T., Santurkar, S., Sastry, G., Schmidt, H., Schnurr, D., Schulman, J., Selsam, D., Sheppard, K., Sherbakov, T., Shieh, J., Shoker, S., Shyam, P., Sidor, S., Sigler, E., Simens, M., Sitkin, J., Slama, K., Sohl, I., Sokolowsky, B., Song, Y., Staudacher, N., Such, F. P., Summers, N., Sutskever, I., Tang, J., Tezak, N., Thompson, M. B., Tillet, P., Tootoonchian, A., Tseng, E., Tuggle, P., Turley, N., Tworek, J., Uribe, J. F. C., Vallone, A., Vijayvergiya, A., Voss, C., Wainwright, C., Wang, J. J., Wang, A., Wang, B., Ward, J., Wei, J., Weinmann, C., Welihinda, A., Welinder, P., Weng, J., Weng, L., Wiethoff, M., Willner, D., Winter, C., Wolrich, S., Wong, H., Workman, L., Wu, S., Wu, J., Wu, M., Xiao, K., Xu, T., Yoo, S., Yu, K., Yuan, Q., Zaremba, W., Zellers, R., Zhang, C., Zhang, M., Zhao, S., Zheng, T., Zhuang, J., Zhuk, W., and Zoph, B. (2024). GPT-4 technical report.

Ovadia, O., Brief, M., Mishaeli, M., and Elisha, O. (2024). Fine-tuning or retrieval? comparing knowledge injection in llms.

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.

Robertson, S. and Zaragoza, H. (2009). The probabilistic relevance framework: BM25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Safavi, T. and Koutra, D. (2021). Relational World Knowledge Representation in Contextual Language Models: A Review. In Moens, M.-F., Huang, X., Specia, L., and Yih, S. W.-t., editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1053–1067, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Seo, M., Baek, J., Thorne, J., and Hwang, S. J. (2024). Retrieval-augmented data augmentation for low-resource domain tasks.

Siriwardhana, S., Weerasekera, R., Wen, E., Kaluarachchi, T., Rana, R., and Nanayakkara, S. (2023). Improving the domain adaptation of retrieval augmented generation (RAG) models for open domain question answering. *Transactions of the Association for Computational Linguistics*, 11:1–17.

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., Bikel, D., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., Fuller, B., Gao, C., Goswami, V., Goyal, N., Hartshorn, A., Hosseini, S., Hou, R., Inan, H., Kardas, M., Kerkez, V., Khabsa, M., Kloumann, I., Korenev, A., Koura, P. S., Lachaux, M.-A., Lavril, T., Lee, J., Liskovich, D., Lu, Y., Mao, Y., Martinet, X., Mihaylov, T., Mishra, P., Molybog, I., Nie, Y., Poulton, A., Reizenstein, J., Rungta, R., Saladi, K., Schelten, A., Silva, R., Smith, E. M., Subramanian, R., Tan, X. E., Tang, B., Taylor, R., Williams, A., Kuan, J. X., Xu, P., Yan, Z., Zarov, I., Zhang, Y., Fan, A., Kambadur, M., Narang, S., Rodriguez, A., Stojnic, R., Edunov, S., and Scialom, T. (2023). Llama 2: Open foundation and fine-tuned chat models.

Wang, Y., Lipka, N., Rossi, R. A., Siu, A., Zhang, R., and Derr, T. (2023). Knowledge graph prompting for multi-document question answering.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E. H., Le, Q. V., and Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Xu, C., Sun, Q., Zheng, K., Geng, X., Zhao, P., Feng, J., Tao, C., and Jiang, D. (2023). Wizardlm: Empowering large language models to follow complex instructions. *arXiv preprint arXiv:2304.12244*.

Zhuang, S., Liu, B., Koopman, B., and Zuccon, G. (2023). Open-source large language models are strong zero-shot query likelihood models for document ranking. In Bouamor, H., Pino, J., and Bali, K., editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8807–8817, Singapore. Association for Computational Linguistics.

A Dataset Construction and Fine-Tuning Settings

To form an IFT dataset from $\mathcal{D}_{\text{train}}$, the technique described in (Cheng et al., 2023) convert raw text extracted from the court decisions into a variety of reading comprehension texts as well as input-output pairs for other common tasks such as summarization, natural-language inference, and more. This procedure yields an IFT dataset with 630 samples consisting of approximately 1M tokens. The fine-tuning procedure uses LoRA (Hu et al., 2022) with an alpha of 32 and dropout of 0.1. The modules tuned with LoRA adapters in each layer are the Q and V matrices. Training proceeds for 10 epochs. An AdamW optimizer (Loshchilov and Hutter, 2018) is used with a learning rate of 2e-05, weight decay of 0.1, betas of $[0.9, 0.95]$ and an epsilon of 1e-05. Gradients are clipped to a norm of 1.0. A small hyper-parameter sweep is performed over three variables. The combinations considered were $\{(8, 4, 4), (8, 4, 2), (16, 4, 4), (8, 2, 2)\}$, where the first entry is rank of the LoRA adaptation matrices, the second is training batch size, and the third is gradient accumulation steps. The model with the best settings $(8, 2, 2)$ was determined based on the lowest validation loss using a 95-5 training-validation split.

To form training examples for embedding models fine-tuning, the raw text of $\mathcal{D}_{\text{train}}$ is partitioned with a chunk size of 1024 and 0 overlap. The default training workflow constructed by LlamaIndex is used.[3] The implementation leverages the Sentence Transformers[4] library and employs the multiple negative ranking loss (Henderson et al., 2017) for training since the dataset only consists of positive pairs. We use a batch size of 10, only 2 epochs of training, and an AdamW optimizer with default settings. Additional default training parameters are found in the `SentenceTransformerTrainingArguments` module[5] of the `sentence_transformers` library. No early stopping or checkpointing is used.

For evaluation, the RAGAs QA generation pipeline used to create the two collections of QA pairs from $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{test}}$ relies on the *Evol-Instruct* method to create diverse questions (Xu et al., 2023). The model `gpt-4o-2024-08-06` is used as the generator and critic LLM during generation. There are 99 QA pairs generated from $\mathcal{D}_{\text{train}}$ and 86 QA pairs created using $\mathcal{D}_{\text{test}}$. Note that, in both cases, the evaluation QA pairs are distinct from the IFT training samples and the context-query pairs used for embedding model fine-tuning.

[3] `https://docs.llamaindex.ai/en/stable/examples/finetuning/embeddings/finetune_embedding/`

[4] `https://sbert.net/docs/package_reference/sentence_transformer/index.html`

[5] `https://sbert.net/docs/package_reference/sentence_transformer/training_args.html#sentencetransformertrainingarguments`