LAR: LLM Assisted Retrieval

Anonymous ACL submission

Abstract

Large language models (LLMs), have demonstrated significant success in natural language understanding and generation tasks. In this work, we propose LAR (Large language model Assisted Retrieval) to harness LLMs towards enhancing the effectiveness of retrieval models, thereby improving the relevance of information retrieval from datasets. Our approach augments a retriever engine by incorporating a subsequent refinement step to the query, utilizing an LLM. This approach showcases the potential of combining retrieval models with LLMs to advance information retrieval systems. We demonstrate the efficacy of LAR through extensive evaluations, specifically showing enhanced performance on the BEIR retrieval benchmark. Furthermore, our methodology exhibits notable improvements on downstream tasks such as question answering, as demonstrated on the NarrativeQA dataset.

1 Introduction

In recent years, the emergence of Large Language Models (LLMs) has revolutionized the landscape of natural language processing tasks. LLMs, such as GPT (Achiam et al., 2023) models, exhibit remarkable capabilities in understanding and generating human-like text. These models leverage large-scale pre-training on diverse text corpora, enabling them to capture intricate linguistic patterns and semantic nuances.

Retrieval models are essential for information retrieval systems, enabling users to locate pertinent documents within vast collections based on their queries. These models optimize the search process by evaluating and ranking the relevance of documents to user queries. Additionally, they are crucial for question answering systems where the corpus is very large.

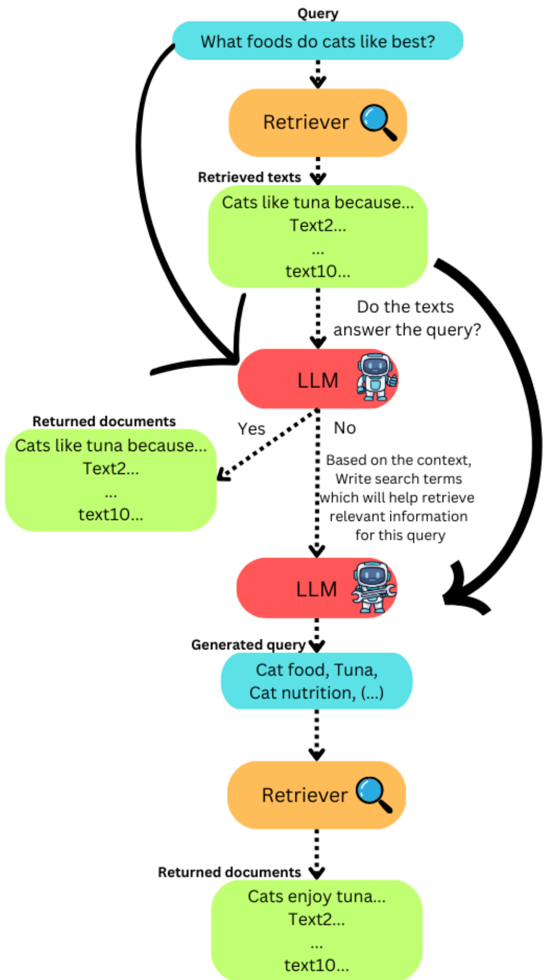

Figure 1: An overview of LAR. The original query is used to gather relevant documents, which are in turn used to prompt the LLM and get a revised query.

Standard retrieval models, such as BM25 (Robertson et al., 2009), leverage statistical methods to rank documents based on term frequency and document length. More advanced models, like Dense Passage Retrieval (Karpukhin et al., 2020), utilize deep learning to understand and match the context of queries and documents via distributed representations of queries and documents. However, a key limitation of these approaches is that they are confined to retrieving documents that are directly related to the given query. Specifically, such methods will struggle with finding documents that require integrating cross-document information. As an example, consider a query about the height of a certain entity John. Assume that there's a document that mentions John is also known as JJ, and another document specifying JJ's height. If there are many documents specifying height, it is unlikely the above retrieval engines will return the one about JJ's height, as it is not sufficiently similar to the query.

Our key observation is that LLMs can "read" retrieved documents, and use these to generate new queries, such that in this process more relevant documents would surface. We implement this idea by introducing a method that uses LLMs to revise queries with retrieval models to return documents given these queries. Importantly, the LLMs generate the queries based on the documents, thus facilitating the use of cross-document information. See Figure 1.

We conducted experiments on the BEIR (Thakur et al., 2021)and ZeroScrolls (Shaham et al., 2023) benchmarks, and have achieved results that outperform all open-source models. The synergy between retrieval models and LLMs presents a promising avenue for advancing information retrieval techniques. By harnessing the contextual understanding and generative prowess of LLMs, it becomes feasible to augment traditional retrieval processes with advanced language understanding capabilities.

Our contributions are:

- We introduce a general method for enhancing the performance of current retrieval systems.
- We investigate the effects of combining LLMs with retrieval models for various tasks.
- We demonstrate competitive results when enhancing existing approaches with LAR.

2 Method

The goal of a document retrieval model can be formally defined as: given a dataset, $\mathcal{D}$, identify and return the most relevant documents in response to a given query, $\mathcal{Q}$. The main challenge lies in efficiently sifting through potentially vast amounts of data to pinpoint documents that best address the query. Our approach involves a two-step iterative process leveraging both a retriever model and a large language model to refine the results.

Our process is illustrated in Figure 1. We begin by employing a retriever model to filter the dataset $\mathcal{D}$, and obtain an initial list of 10 documents. From the list of documents retrieved in the first step, we select the top k documents. These documents, along with the query $\mathcal{Q}$ are then presented to the LLM. The LLM assesses whether these documents contain answers pertinent to the query in a zero-shot manner. If the LLM confirms that the documents are relevant, these documents are used as the final answer set. If the LLM determines that the documents are not sufficiently relevant, we prompt it to generate a set of more appropriate search terms. These updated terms form a revised query. The retriever model is then employed again using the updated query. The documents retrieved in this iteration are taken as the final answer set.

Finally, we have applied a reranking model to the result, demonstrating that LAR can be combined with existing approaches to enhance their performance.

3 Experiments

In what follows, we provide details about the datasets and evaluation protocol, and implementation details.

3.1 BEIR Benchmark Evaluation

We evaluated our approach on the BEIR[1] benchmark datasets (Thakur et al., 2021). BEIR consists of 18 datasets containing information retrieval tasks. We used the standard test sets for all evaluations. Initially, we use Pyserini's (Lin et al., 2021) flat indexes to retrieve 10 documents for each query using the BM25 algorithm with default parameters (k1=0.9, b=0.4). These documents, along with the query, were then provided as input into GPT4o.

The task of the LLM is to assess whether the retrieved documents sufficiently addressed the query by answering the question: "Do the texts contain the answer to the query? Answer in 'Yes' or 'No'."

If the LLM responded "No", indicating that the retrieved documents were insufficient, we prompted the LLM again to understand what additional information was needed. Specifically, the LLM was asked: "Use these texts and the query

[1] We did not test LAR on the Quora and CQADupstack datasets because our approach is tailored for a query and a related corpus containing information about it. These datasets consist of query duplications where the task is to find the most similar question, with each "document" being a question. Instead, we present the average BM25 result or the average InPars result.

provided to better understand what information is missing from the texts to answer the query, and provide search terms that will help search a larger text for that missing information." The LLM-generated search terms were then used to refine the query and retrieve a new set of documents using BM25-flat.

The final set of documents obtained through this refined retrieval process was evaluated using the nDCG@10 score.

As a final optional step, we took an existing reranking method, InPars-v2 (Jeronymo et al., 2023), and incorporated it on top of our method, in order to assess if LAR can be combined with existing methods for superior results. InPars-v2 used data generated from the BEIR datasets in order to fine tune a Monot5 (Nogueira et al., 2020) model to be used as a reranker for the final 1000 results retrieved using BM25-flat. We used these reranker models as the final step in our process. Once the final result of the retrieved documents was returned from BM25, we rereanked the results.

3.2 Open Domain QA

We also conducted experiments using the NarrativeQA (Kočiskỳ et al., 2018) dataset from the ZeroScrolls benchmark (Shaham et al., 2023), in order to evaluate the improvement in open-domain question answering. The ZeroScroll subset is a subset of the NarrativeQA dataset that contains 500 datapoints. We have used that instead of the full narrativeQA dataset for budgetary reasons. Here, the objective is to extract a precise answer from a lengthy document in response to a given query. Out of the ZeroScrolls, we focused only on NarrativeQA since it is the only one that both has queries, and has texts long enough to make retrieval useful (as opposed to just using the entire text as context).

The original corpus was preprocessed by segmenting it into chunks of approximately 500 characters each, ending at the first period after the 500th character. These chunks were treated as individual documents for the retrieval process.

Following the preprocessing, we adhered to the previously described retrieval steps involving the LLM for generating new search terms if the initial documents were deemed insufficient, but using a Dense Passage Retrieval model, Contriever (Izacard et al., 2021) for the retrieval of documents, and gpt-4-1106-preview as the LLM.

In the final retrieval step, the documents selected based on LLM guidance were used as context for question answering. Answers were evaluated using F1 score, as is standard for this dataset. It is important to note that our approach utilized approximately 1700 tokens as context, in contrast to the ZeroScrolls baseline which used 8000 tokens.

3.3 Incorporating InPars-V2 Rerankers

For the BEIR benchmark, we also explored the use of a reranker, InPars-v2, and incorporated it on top of LAR. InPars-v2 used data generated from the BEIR datasets in order to fine tune a Monot5 model to be used as a reranker for the final results retrieved from BM25-flat. We used these reranker models as the final step in our process. Once the final result of retrieved documents was returned from BM25, we reranked the results before evaluating them.

4 Results

Table 1 presents the BEIR results for BM25, BM25 enhanced by the LLM, BM25 followed by an InParsV2 reranker, and BM25 enhanced by the LLM and followed by an InParsV2 reranker. Our approach shows statistically significant (paired t-test. $p = 0.039$) and consistent improvements over InParsV2. In addition, LAR shows significant improvement when applied over BM25 without the use of a reranker. These results beat the current best open source method (Jeronymo et al., 2023), while still being competitive with the unpublished current state of the art, reported on the BEIR benchmark leaderboard.

Table 2 presents the results on the ZeroScrolls subset of the NarrativeQA dataset. We demonstrate the efficacy of LAR when employing a dense passage retrieval model, Contriever. We achieve superior results to the ZeroScrolls baseline despite using only roughly 1700 tokens in comparison to the baseline's 8000.

5 Related Work

Iterative retrieval has been explored before in different contexts. (Trivedi et al., 2022) uses chain-of-thought to guide the retrieval process and refines the CoT with the obtained retrieval results. They differ from our approach because while they utilize the LLM for query enhancement, we also use the LLM for determining whether the current context is sufficient, thus avoiding model hallucinations in QA. Peng et al. (2023) enhances LLM responses by grounding them in external knowledge and refining prompts with utility function feedback, but this external knowledge must be stored in a task specific

Dataset	BM25 Enhanced by LLM + InParsV2	BM25 + InParsV2	BM25 + Enhanced by LLM	BM25
AVG	**0.55**	**0.545**	**0.443**	**0.424**
NFCorpus	0.405	0.385	0.339	0.321
Arguana	0.372	0.369	0.404	0.397
Trec-covid	0.848	0.846	0.609	0.594
Touche-2020	0.291	0.291	0.460	0.442
Dbpedia-entity	0.505	0.498	0.351	0.318
Scidocs	0.208	0.208	0.158	0.149
Climate-FEVER	0.324	0.323	0.190	0.165
Scifact	0.770	0.774	0.715	0.678
Fiqa	0.516	0.509	0.251	0.236
Fever	0.860	0.872	0.640	0.651
Nq	0.653	0.638	0.354	0.305
Hotpotqa	0.795	0.791	0.654	0.633
Robust04	0.656	0.632	0.461	0.408
Trec-news	0.493	0.49	0.447	0.395
Signal1m	0.312	0.308	0.322	0.330
Bioasq	0.594	0.595	0.523	0.522
CQADupstack	0.448	0.448	0.302	0.302
Quora	0.845	0.845	0.789	0.789

Table 1: nDCG@10 on BEIR. Improvement over the baseline is signifcant (paired t-test. $p = 0.039$). The CQADupstack and Quora values are placeholders since we did not run experiments on these datasets because they focus on question duplication rather than document retrieval to answer a query.

Dataset	Contriever + Enhanced by LLM	Contriever	ZeroScrolls baseline ($\sim$8000 tokens)
NarrativeQA	33.7	33.1	27.6

Table 2: F1 on NarrativeQA. The ZeroScrolls baseline used 8000 tokens for their evaluation, while LAR used only $\sim$1700 tokens.

database. Zemlyanskiy et al. (2022) retrieve exemplars with outputs similar to a preliminary output generated by the LLM. (Yu et al., 2023) uses a generated output to retrieve relevant context used for output refinement, while we prompt the LLM to, when necessary, refine the input (query) by using retrieved documents (context).

Retrieval-Augmented Generation (RAG) (Guu et al., 2020; Lewis et al., 2020) presented a technique that enhances language models (LMs) by incorporating relevant text passages retrieved from external sources into their input space. This approach has been shown to significantly boost performance in knowledge-intensive tasks, both when fine-tuned and when used with pre-trained LMs, however it does not boost the retrieval process itself. (Gao et al., 2023) used reflection tokens to adaptively retrieve passages, determining the best moment for retrieval. (Luo et al., 2023) fine-tunes a language model by prepending a fixed number of relevant retrieved passages to the input. Jiang et al. (2023) adaptively retrieves passages to assist generation based on the confidence of the previously generated tokens.

6 Conclusion

In this work, we presented a general approach for utilizing LLMs to enhance information retrieval systems and performance on downstream tasks. Our results demonstrate that LAR can be used to enhance retrieval processes based on BM25, DPR and reranker models, resulting in improved retrieval quality. Our evaluation demonstrates competitive performance with strong baselines on BEIR and NarrativeQA. It is likely these results can be improved further by introducing more elaborate iterative procedures that take into account information gathered from previous retrievals.

7 Limitations

Despite the promising results described here, few limitations need to be acknowledged. First, LAR requires one or two calls to an LLM. This reliance can introduce significant computational costs and time delays, especially when dealing with large-scale datasets or real-time applications. Our model's retrieval performance relies on the corpus containing relevant information about the query, limiting its effectiveness in scenarios where this is not the case, such as duplicated question retrieval, as exemplified by datasets like "Quora". These are general limitations on these kinds of models.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*.

Vitor Jeronymo, Luiz Bonifacio, Hugo Abonizio, Marzieh Fadaee, Roberto Lotufo, Jakub Zavrel, and Rodrigo Nogueira. 2023. Inpars-v2: Large language models as efficient dataset generators for information retrieval. *arXiv preprint arXiv:2301.01820*.

Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation. *arXiv preprint arXiv:2305.06983*.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

Tomáš Kočiskỳ, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The narrativeqa reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: An easy-to-use python toolkit to support replicable ir research with sparse and dense representations. *arXiv preprint arXiv:2102.10073*.

Hongyin Luo, Yung-Sung Chuang, Yuan Gong, Tianhua Zhang, Yoon Kim, Xixin Wu, Danny Fox, Helen Meng, and James Glass. 2023. Sail: Search-augmented instruction learning. *arXiv preprint arXiv:2305.15225*.

Rodrigo Nogueira, Zhiying Jiang, and Jimmy Lin. 2020. Document ranking with a pretrained sequence-to-sequence model. *arXiv preprint arXiv:2003.06713*.

Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou Yu, Weizhu Chen, et al. 2023. Check your facts and try again: Improving large language models with external knowledge and automated feedback. *arXiv preprint arXiv:2302.12813*.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Uri Shaham, Maor Ivgi, Avia Efrat, Jonathan Berant, and Omer Levy. 2023. Zeroscrolls: A zero-shot benchmark for long text understanding. *arXiv preprint arXiv:2305.14196*.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663*.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. *arXiv preprint arXiv:2212.10509*.

Wenhao Yu, Zhihan Zhang, Zhenwen Liang, Meng Jiang, and Ashish Sabharwal. 2023. Improving language models via plug-and-play retrieval feedback. *arXiv preprint arXiv:2305.14002*.

Yury Zemlyanskiy, Michiel de Jong, Joshua Ainslie, Panupong Pasupat, Peter Shaw, Linlu Qiu, Sumit Sanghai, and Fei Sha. 2022. Generate-and-retrieve: Use your predictions to improve retrieval for semantic parsing. *arXiv preprint arXiv:2209.14899*.