Aligning Embedding with LLM by Citation Enhanced Generation

Anonymous ACL submission

Abstract

Large Language Models (LLMs) have demonstrated remarkable general intelligence but still struggle with hallucination problems. Retrieval Augmented Generation (RAG) addresses it by incorporating external knowledge sources. However, a critical challenge in RAG systems is the misalignment between embedding-based retriever and LLM generator. This paper introduces a novel approach to align the embedding model with LLM through Citation Enhanced Generation (CEG). Our method leverages citation information from LLM outputs to create positive and negative training samples for embedding model fine-tuning. This method incorporates LLM feedback into embedding model training, thereby achieving alignment between them. Experimental results demonstrate significant improvements in RAG performance across multiple datasets, with particularly notable gains in specialized domains.

1 Introduction

In recent years, Large Language Models (LLMs) have achieved remarkable progress and demonstrated powerful capabilities across various natural language processing tasks (Zhao et al., 2025). However, LLMs still face hallucination issues in certain scenarios, where they generate answers that contradict facts (Ji et al., 2023; Bang et al., 2023). This phenomenon can be attributed to the inherent limitations of static parameters, which only internalize knowledge encountered during the training phase and lack the capability to dynamically update in response to emerging world knowledge.

To address this challenge, Retrieval Augmented Generation (RAG) methods have emerged (Gao et al., 2024). RAG enhances the accuracy of LLM responses by incorporating information from external knowledge bases, thereby mitigating hallucination problems. Embedding models (Nie et al., 2025) play a crucial role in RAG systems, as they retrieve documents relevant to input queries. These embedding models encode text into vector representations, and high-quality representations enable various downstream tasks such as classification and retrieval. As a key component of RAG, embedding model directly impacts the quality of the final generated answers.

However, existing RAG methods lack effective alignment between embedding model and LLM. These components differ in knowledge representation and comprehension. This misalignment can cause problems: documents that appear similar in the embedding model's representation space may not provide substantial support for LLM-generated answers. Therefore, how to effectively align embedding with LLM to eliminate this gap becomes a significant challenge in RAG methods. Some recent works attempt to address this issue by obtaining signals from LLM outputs to train embedding. LLM-Embedder (Zhang et al., 2024) introduces a new reward formulation, namely rank-aware reward. It utilizes the ranking position of expected outputs among N sampled outputs from the LLM, which leads to computation of reward from the LLM's feedback.

In this work, we propose a simple and intuitive method to align embedding model with LLM through Citation Enhanced Generation (CEG). CEG enables LLMs to generate text with citations, improving factual accuracy and verifiability (Gao et al., 2023; Li et al., 2024). Our approach leverages citation information from CEG to distinguish between documents that contribute to response correctness and those do not. Based on this distinction, we construct positive and negative samples for each question and use these samples to fine-tune embedding models. Through this process, we enable embedding models to preferentially retrieve documents that provide factual support for LLM answers, rather than documents that are merely semantically similar to the questions in their rep-

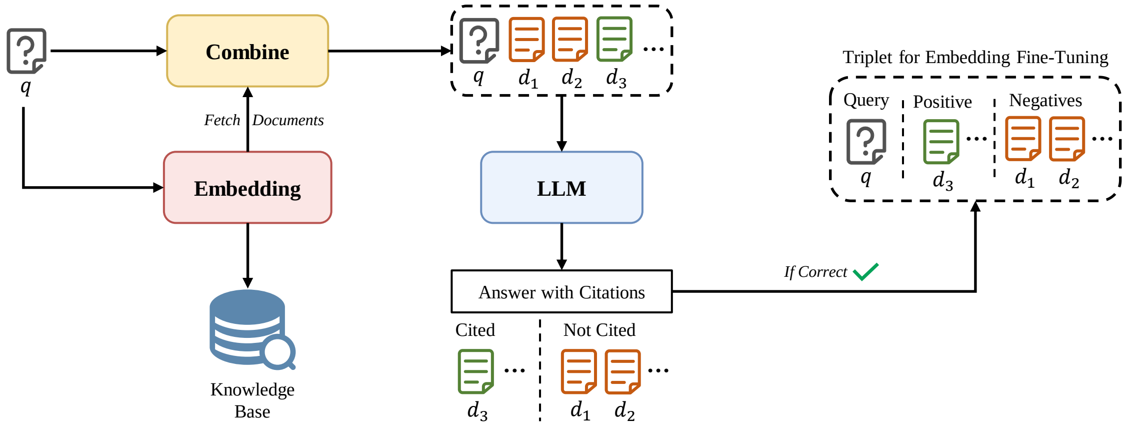

Figure 1: The illustration of our method. Citation information from LLM output is used to construct positive and hard negative samples for embedding model fine-tuning.

resentation space. Our method constructs positive-negative sample datasets containing LLM feedback signals at low cost and fine-tunes embedding model to align them with LLM.

In summary, the main contributions of our work can be summarized as follows:

- To the best of our knowledge, we are the first to propose using CEG to align embedding models with LLMs.
- We present a simple but effective method that constructs positive and negative sample datasets using citation information generated by CEG, and leverages them to fine-tune embedding for alignment with LLM.
- Experimental results show the effectiveness of our approach in aligning embedding models with LLMs, with particularly significant improvements in specialized domains.

2 Methodology

2.1 Formulation

Retrieval Augmented Generation mitigates hallucination in LLM by incorporating external knowledge into the generation process. Given an input question q, the retriever first fetches k relevant documents $\mathcal{D} = \{d_1, d_2, \cdots, d_k\}$ from a knowledge base $\mathcal{KB}$:

$$\mathcal{D} = \text{Retriever}(q, k, \mathcal{KB}). \quad (1)$$

The retriever can employ sparse methods like BM25 (Robertson and Zaragoza, 2009) or TF-IDF (Salton and Buckley, 1988). It can also use dense retrieval methods based on embedding models. Due to the powerful semantic representation capabilities of embedding models, dense retrieval methods are often the preferred solution for RAG. The retrieved documents are combined with the question q as context. Together, they serve as input to prompt the LLM, which generates the final answer $\mathcal{S}$:

$$\mathcal{S} = \text{LLM}(q, \mathcal{D}). \quad (2)$$

Citation Enhanced Generation serves as a further enhancement to RAG. When generating answers, CEG explicitly cites relevant documents. This enables answer tracing and verification, thereby increasing answer credibility. After RAG processing, the LLM output $\mathcal{S}$ can be segmented into n statements $s_1, s_2, ..., s_n$. Each statement s_i may optionally cites a list of passages $\mathcal{C}_i = \{c_{i,1}, c_{i,2}, ...\}$, where $c_{i,j} \in \mathcal{D}$. Here, $\mathcal{D}$ represents the set of relevant documents retrieved by the retriever, and $c_{i,j}$ provides factual support for statement s_i. In this work, we use symbol such as `[1][2]` to mark $\mathcal{C}_i$. In summary, the CEG generation process can be represented as:

$$\mathcal{S} = s_1\mathcal{C}_1, s_2\mathcal{C}_2, ..., s_n\mathcal{C}_n = \text{LLM}(q, \mathcal{D}). \quad (3)$$

2.2 Align Embedding with LLM

Embedding models are typically fine-tuned through contrastive learning. This fine-tuning paradigm requires constructing numerous positive and negative samples. These samples help the model learn text similarities and differences. We propose using information generated by CEG to construct datasets of positive and negative samples for fine-tuning embedding models. This method eliminates the cost of manual data annotation. It also introduces LLM preferences into the embedding training data. This further aligns the embedding model with the LLM. The overall process of our proposed method is illustrated in Figure 1.

Specifically, for a given question q, we execute the CEG process using the embedding model and LLM according to equations 1 and equations 3. This produces a final answer $\mathcal{S}$. We then determine whether the answer $\mathcal{S}$ is correct for question q. If correct, documents $c_{i,j}$ cited in $\mathcal{S}$ are considered positive samples for q and documents in $\mathcal{D}$ that are not cited are treated as hard negative samples. If incorrect, no positive or negative samples are generated from this CEG process.

We iterate through this process on the training set of a QA dataset. We gradually collect triplets of positive and negative samples $(q, p, \mathcal{N})$. Here, q is the question from the QA sample. $p \in \bigcup_{i=1}^{n} \mathcal{C}_i$ is a positive sample for q. $\mathcal{N} = \{d \in \mathcal{D} | d \notin \bigcup_{i=1}^{n} \mathcal{C}_i\}$ is a set of hard negative samples for q. Each QA data point can produce multiple such triplets. These triplets have different positive samples but share the same set of hard negative samples.

All triplets collectively form a fine-tuning dataset $\mathcal{T}$. We use $\mathcal{T}$ to fine-tune the embedding model. The loss function can be expressed as:

$$\mathcal{L} = -\log \frac{\exp(\mathrm{s}(q,p))}{\exp(\mathrm{s}(q,p)) + \sum_{d \in \mathcal{N}} \exp(\mathrm{s}(q,d))}, \tag{4}$$

where $\mathrm{s}(x, y)$ represents the similarity between x and y. We use cosine similarity to calculate this similarity.

3 Experiments

3.1 Settings

3.1.1 Datasets

To facilitate the evaluation of answer correctness, we select three multiple-choice QA datasets for our experiments. **1) MedQA** (Jin et al., 2021) is a medical domain multiple-choice QA dataset collected from professional medical licensing exams. We use the English version of MedQA and utilized its built-in knowledge base for retrieval. **2) OpenBookQA** (Mihaylov et al., 2018) is a multiple-choice QA dataset simulating open-book exams in the scientific domain. We collect fact fields from all data samples and combined them with the provided commonsense fact corpus to construct a knowledge base for retrieval. **3) QASC** (Khot et al., 2020) is a scientific domain multiple-choice QA dataset focusing on sentence composition reasoning. QASC has a fact corpus containing 17 million entries. To reduce the complexity of knowledge base vectorization, we randomly sample a subset of facts and combined them with fact fields and composition fact fields from the dataset samples, creating a knowledge base of 200,000 entries for retrieval.

For each dataset, we apply proposed method to construct positive and negative samples from the training set and then fine-tuned the embedding model. After fine-tuning, we evaluate the impact of the fine-tuned embedding models on answer correctness using the test set. We use accuracy as the evaluation metric. The statistics for each dataset and constructed samples are shown in Table 3 and Table 4.

3.1.2 Baselines

We implement several retrieval methods and embedding models as baselines for comparative experiments: BM25 (Robertson and Zaragoza, 2009), BGE-large-en-v1.5 (Xiao et al., 2024), E5-large-v2 (Wang et al., 2024), BGE-M3 (Chen et al., 2024), and LLM-Embedder (Zhang et al., 2024). BM25 serves as a classic sparse retrieval method. BGE-large-en-v1.5 and E5-large-v2 are two BERT-based embedding models that support English. BGE-M3 is a multilingual embedding model that supports long-text and hybrid retrieval. LLM-Embedder is an embedding model trained using LLM reward signals, supporting English. For each dataset, we employ Qwen2.5-3B-Instruct and Qwen2.5-7B-Instruct (Yang et al., 2024) as generators paired with different retrievers.

3.1.3 Implementation

We select BGE-base-en-v1.5 (Xiao et al., 2024) as our base embedding model. For each dataset, we construct positive and negative samples by pairing with the corresponding LLM. We then fine-tune checkpoints accordingly. We employ simple prompt engineering to guide the CEG process. The prompt template is shown in Figure 2. We implement embedding model fine-tuning using the SentenceTransformers (Reimers and Gurevych, 2019) library. For the BM25 algorithm, we utilize the BM25s (Lù, 2024) library. We use Flat index from faiss (Johnson et al., 2021) for dense retrieval. We deploy embedding model and LLM via the vLLM (Kwon et al., 2023) library to accelerate the data

Generator	Retriever	MedQA	OBQA	QASC
Qwen2.5-3B-Instruct	BM25	43.36	73.60	90.17
	BGE-large-en-v1.5 (335M)	47.53	74.20	91.58
	E5-large-v2 (335M)	40.30	70.00	79.48
	BGE-M3 (568M)	41.95	76.00	89.85
	LLM-Embedder (110M)	45.56	72.60	89.96
	Ours (110M)	**49.80**	**76.60**	**92.19**
Qwen2.5-7B-Instruct	BM25	52.71	85.80	96.11
	BGE-large-en-v1.5 (335M)	55.93	86.00	97.52
	E5-large-v2 (335M)	53.89	82.80	89.31
	BGE-m3 (568M)	53.65	85.20	97.30
	LLM-Embedder (110M)	54.75	86.20	96.54
	Ours (110M)	**60.17**	**86.80**	**97.84**

Table 1: Evaluation results. The metric is accuracy (%). The best results are in **bold**. The content in parentheses after the embedding name indicate the number of parameters.

Generator	Retriever	MedQA	OBQA	QASC
Qwen2.5-3B-Instruct	BGE-base-en-v1.5	45.40	73.60	89.85
	w/ IBN	46.98	75.60	91.58
	w/ HN	**49.80**	**76.60**	**92.19**
Qwen2.5-7B-Instruct	BGE-base-en-v1.5	58.21	85.80	96.87
	w/ IBN	57.74	86.20	97.30
	w/ HN	**60.17**	**86.80**	**97.84**

Table 2: Ablation study of our method with different negative strategy. The metric is accuracy (%).

synthesis and evaluation process. We set the number of documents to be retrieved to 10. We set the LLM temperature to 0 to ensure output stability and reproducibility.

3.2 Main Results

We present our comparative experimental results in Table 1. Our method outperforms all baselines across all datasets. The embedding models fine-tuned with our method achieve better performance even when compared to larger embedding models. At the same model scale, LLM-Embedder uses a similar concept of obtaining reward signals from LLM to fine-tune embedding models. However, our method achieves higher accuracy. This demonstrates the superior effectiveness of our approach in aligning embeddings with LLM. Additionally, we observe that our method shows more significant improvements on specific domain datasets like MedQA. This phenomenon indicates that in specific domains, the knowledge gap between LLM and embedding models is larger. Therefore, alignment becomes more crucial in specific domains.

3.3 Ablation Study

We conduct ablation study to evaluate different negative sample strategy. **1) In-batch negatives (IBN)**: This method only uses the positive samples constructed in our approach. During training, it employs positive samples from other instances in the same batch as negative samples for the current instance. **2) Hard negatives (HN)**: This is our proposed method. It fully utilizes the constructed positive and negative samples to train the embedding model. The results of the ablation experiments are shown in Table 2. We observe that the HN method brings more significant improvements. This indicates the rationality of treating documents retrieved by the embedding model but not cited by the LLM as hard negative samples. These documents are considered relevant to the question by the embedding model. However, they do not actually provide support for the answers generated by the LLM. Therefore, using them as hard negative samples helps the embedding model better learn this distinction. This strategy effectively aligns the embedding model with the LLM.

4 Conclusion

In this work, we explore how to addresses the misalignment between embedding and LLM in RAG systems. By leveraging citation signals from LLM outputs to construct positive and negative samples, we establish an effective feedback that enables embedding to better align with LLM. Our experimental results demonstrate the effectiveness of proposed method on several datasets. This highlights the potential of using CEG to enhance the alignment between embedding and LLM.

Limitations

While our method demonstrates promising results, several limitations remain. Our approach relies on the citation quality of LLMs, which may not always be accurate or comprehensive. If LLMs make incorrect citations, these errors could propagate into embedding model training. Besides, our experiments are based on in-domain observations. Whether the performance improvements achieved by the proposed method within domains can be generalized requires further study.

References

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, and Pascale Fung. 2023. A multitask, multilingual, multimodal evaluation of ChatGPT on reasoning, hallucination, and interactivity. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 675–718, Nusa Dua, Bali. Association for Computational Linguistics.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *Preprint*, arXiv:2402.03216.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023. Enabling large language models to generate text with citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6465–6488, Singapore. Association for Computational Linguistics.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey. *Preprint*, arXiv:2312.10997.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2021. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14).

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2021. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547.

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. 2020. Qasc: A dataset for question answering via sentence composition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8082–8090.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, SOSP '23, page 611–626, New York, NY, USA. Association for Computing Machinery.

Weitao Li, Junkai Li, Weizhi Ma, and Yang Liu. 2024. Citation-enhanced generation for LLM-based chatbots. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1451–1466, Bangkok, Thailand. Association for Computational Linguistics.

Xing Han Lù. 2024. Bm25s: Orders of magnitude faster lexical search via eager sparse scoring. *Preprint*, arXiv:2407.03618.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.

Zhijie Nie, Zhangchi Feng, Mingxin Li, Cunwang Zhang, Yanzhao Zhang, Dingkun Long, and Richong Zhang. 2025. When text embedding meets large language model: A comprehensive survey. *Preprint*, arXiv:2412.09165.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

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

Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2024. Text embeddings by weakly-supervised contrastive pre-training. *Preprint*, arXiv:2212.03533.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. 2024. C-pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 641–649, New York, NY, USA. Association for Computing Machinery.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 43 others. 2024. Qwen2 technical report. *Preprint*, arXiv:2407.10671.

Peitian Zhang, Zheng Liu, Shitao Xiao, Zhicheng Dou, and Jian-Yun Nie. 2024. A multi-task embedder for retrieval augmented LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3537–3553, Bangkok, Thailand. Association for Computational Linguistics.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, and 3 others. 2025. A survey of large language models. *Preprint*, arXiv:2303.18223.

A Dataset Details

Dataset	**# Corpus**	**# Train Set**	**# Test Set**
MedQA	213,330	10,178	1,273
OBQA	6,492	4,957	500
QASC	200,000	8,134	926

Table 3: The statistics of datasets.

Dataset	**# Constructed Samples**	
	Qwen2.5-3B-Instruct	Qwen2.5-7B-Instruct
MedQA	5,727	16,267
OBQA	20,499	17,194
QASC	42,666	38,715

Table 4: The statistics of constructed samples.

B Prompt Template

```
Instruction: Answer multiple-choice questions
based on searched documents. You are required to
give a detailed analysis that includes citations to
relevant documents. When citing documents, use
numbers such as [1][2][3]. Remember to only cite
documents that are helpful to the question. After the
analysis, give the best answer option label without
adding any extra content.

Documents:
{document_list}

Question: {question}
Options:
{options}

Next, give your answer. Format is:
Analysis: {{your analysis with citations}}
Choice: {{option label}}
```

Figure 2: Prompt template for CEG.