Advancing Academic Knowledge Retrieval via LLM-enhanced Representation Similarity Fusion: The 2nd Place of KDD Cup 2024 OAG-Challenge AQA

Wei Dai
Robo Space
loveispdvd@gmail.com

Peng Fu
Robo Space
fupeng@hotmail.com

Chunjing Gan
Ant Group
cuibing.gcj@antgroup.com

ABSTRACT

In an era marked by robust technological growth and swift information renewal, furnishing researchers and the populace with top-tier, avant-garde academic insights spanning various domains has become an urgent necessity. The KDD Cup 2024 AQA Challenge is geared towards advancing retrieval models to identify pertinent academic terminologies from suitable papers for scientific inquiries. This paper introduces the LLM-KnowSimFuser proposed by *Robo Space*, which wins the 2nd place in the competition. With inspirations drawed from the superior performance of LLMs on multiple tasks, after careful analysis of the provided datasets, we firstly perform fine-tuning and inference using LLM-enhanced pre-trained retrieval models to introduce the tremendous language understanding and open-domain knowledge of LLMs into this task, followed by a weighted fusion based on the similarity matrix derived from the inference results. Finally, experiments conducted on the competition datasets show the superiority of our proposal, which achieved a score of 0.20726 on the final leaderboard.

KEYWORDS

Information Retrieval, Ensemble Learning, KDD Cup 2024

ACM Reference Format:
Wei Dai, Peng Fu, and Chunjing Gan. 2024. Advancing Academic Knowledge Retrieval via LLM-enhanced Representation Similarity Fusion: The 2nd Place of KDD Cup 2024 OAG-Challenge AQA. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (KDD'24).* ACM, New York, NY, USA, 4 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

The overarching aim of scholarly data mining is to enhance our comprehension of the progression, essence, and direction of science. It possesses the capability to unveil substantial scientific, technological, and educational worth. In an age of vigorous technological expansion and rapid informational refreshment, equipping scholars and the general public with premier, cutting-edge academic knowledge across diverse disciplines is now an imperative demand. The 2024 KDD Cup AQA competition is oriented toward enhancing retrieval algorithms with the aim to pinpoint relevant academic publications for scientific queries[10]. Inspired by the remarkable achievements of large language models (LLMs) such as ChatGPT [8], GPT4 [9] in a variety of tasks due to their marvelous capability in language comprehension and generation, in this paper, we introduces the LLM-KnowSimFuser solution proposed by *Robo Space*, which incorporates the tremendous language understanding and open-domain knowledge of LLMs in this solution and wins the 2nd place in the competition (achieved a score of 0.20726 on the final leaderboard) and organizes this technical report as follows:

- First, we outline the task objectives and present the statistics of the given datasets in detail.
- Subsequently, we introduce our data processing flow and process for fine-tuning and inference on LLM-enhanced pre-trained retrieval models with a carefully designed similarity fusion mechanism.
- Finally, we conduct comprehensive ablation study and parameter analysis experiments on the competition datasets, which demonstrate the effectiveness and superiority of our proposal.

2 DATASETS

The KDD Cup 2024 Academic Question Answering (AQA) Challenge is centered on tackling an academic retrieval problem. This endeavor employs a dataset that is systematically organized into two primary components: queries and documents. Queries embody academic questions, each structured with a concise title and an elaborative body that delineates the question's specifics. Documents, on the other hand, represent academic papers, each comprising a descriptive title and an informative abstract that encapsulates the paper's core contributions.

All participants are required to navigate through two phases of the competition, with the latter building upon the former with enhanced complexity. The initial phase challenge contenders with a defined set of queries and documents, requiring the identification of the most pertinent documents for each query. Progressing to the second phase, the test set expands and the document collection significantly grows, escalating the task's intricacy while the core objective of discerning relevancy persists. We list the details of the statistics and objectives in Table 1.

3 METHODOLOGY

Our approach is composed of three main components:

- **Embedding Extraction using Pre-trained Models:** We employ several distinct LLM-enhanced pre-trained models to extract embeddings separately for queries and documents.

KDD'24, August, 2024, Barcelona, Spain

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

Phase	Training Set	Test Set	Paper Collection	Objective
1	8,757	2,919	395,812	Top 20 IDs per query
2	Same as Phase 1	5919 (+3,000)	466,387 (+70,575)	Same as Phase 1

Table 1: Competition phases statistics and objectives summary.

- **Tuning a Pre-trained Model:** Among the pre-trained models, one is fine-tuned. We then use this tuned model to extract embeddings for both queries and documents.
- **Similarity Matrix Computation and Fusion:** Embeddings from the above models (five in total, including the tuned one) are used to compute similarity matrices between queries and documents. We then fuse and rank these five similarity matrices to improve relevance assessment.

3.1 Utilization of Pre-trained Retrieval Models

We utilize four pre-trained models: NV-Embed-v1[6], SFR-Embedding-Mistral[1], GritLM-7B[7], and Linq-Embed-Mistral[5]. All these models are based on the Mistral framework [3], which *excel in capturing rich semantic relationships and contextual nuances with additional open-domain knowledge, leading to more accurate and relevant search results compared to traditional retrieval models*[2]. Besides, they share similar methods for prompt construction and embedding extraction. Notably, the GritLM-7B model employs mean pooling by default, while the other three models utilize last token pooling. Although it is feasible to use different pooling techniques with GritLM, we adhere to mean pooling to remain consistent with the convention established during pre-training.

For document embedding extraction, embeddings can be directly obtained without additional prompt words. However, for queries, which are typically short and sparse in content, it is crucial to differentiate them from documents in a retrieval setting. Therefore, we experimented with various instructions and tags as prompts to enhance query embeddings, where we present the results of different configurations of tags and instructions in Section 4.

3.2 Supervised Fine-Tuning of Retrieval Models

Among the evaluated models, the SFR-Embedding-Mistral model proved to be the most suitable candidate for fine-tuning due to its simplicity and inherent flexibility. We opt to use the Tevatron [1] framework in conjunction with the Low-Rank Adaptation (LoRA) [2] method to optimize the model's performance. For the fine-tuning process, we employed a comprehensive dataset comprising queries and academic papers in the training set, ensuring that the model can be well-adapted to handle the specific academic retrieval tasks with high accuracy.

To achieve this, we meticulously configure several key LoRA parameters. Specifically, we set the scaling factor to 64 and applied a dropout rate of 0.1 to prevent overfitting by randomly deactivating a fraction of neurons during training. Additionally, we define the rank of the low-rank matrices used for adaptation as 8. These configurations are chosen to strike a balance between model complexity and performance. Notably, the model is fine-tuned for only one epoch. This decision is based on empirical evidence indicating that additional epochs of training led to a decline in performance, likely due to overfitting. By limiting the fine-tuning process to a single epoch, we are able to maintain the model's optimal performance and generalization capabilities without compromising its effectiveness.

3.3 Similarity Fusion of Pretrained Models

In this section, we describe the process of integrating similarity matrices derived from multiple models to achieve a unified and robust retrieval outcome. We utilize four pretrained models and one fine-tuned model, as detailed in Sections 3.1 and 3.2. The following steps will outline the detailed procedure for computing, normalizing, and fusing the similarity matrices.

Firstly, we compute the similarity matrices for the embeddings of queries and documents from each model independently. This involves measuring the similarity between the query embeddings and document embeddings generated by each model. We employ Faiss [4], an efficient library that leverages GPU acceleration, to expedite these similarity calculations. The use of GPU acceleration significantly enhances the computation speed, making it feasible to handle large-scale data efficiently. Next, to ensure comparability across different models, we normalize the similarity matrices for each model on a per-query basis. Normalizing the similarity matrices ensures that the scores from different models are on a uniform scale, which is crucial for fair and effective fusion. After normalization, we perform a weighted fusion of the similarity matrices. Each model's normalized similarity matrix is assigned a weight that reflects its relative importance or performance, which combines the strengths of the individual models, leveraging their diverse perspectives to improve the robustness and accuracy of the similarity measurements. Finally, based on the aggregated similarity scores, we rank the documents for each query. We identify and select the top 20 documents with the highest similarity scores as the final results for submission. This selection process ensures that the most relevant documents, as determined by the combined insights of multiple models, are presented as the output.

4 EXPERIMENTS

4.1 Experimental Setup and Reproducibility

All experiments were conducted on a machine equipped with an Intel(R) Xeon(R) Silver 4210 CPU @ 2.20GHz, 128GB of RAM, and an NVIDIA RTX A6000 GPU with 48GB of memory. We fine-tune SFR-Embedding-Mistral model using LoRA on specific modules (q_proj, k_proj, v_proj, o_proj, down_proj, up_proj, and gate_proj) with a learning rate of 1e-4, per-device batch size of 8, 1 epoch of

[1] https://blog.salesforceairesearch.com/sfr-embedded-mistral/
[2] During experiments, we incorporate traditional retrieval models such as BGE when testing, however the outcome is not promising which demonstrates the superiority of LLM-enhanced retrieval models in such specific domains.

training, query and passage lengths limited to 32 and 156 tokens, respectively. To promote reproducibility, our source code is publicly available on GitHub[3], providing comprehensive guidance on the operational processes. Detailed instructions on how to run the code can be found in the README.md file within the repository. Additionally, specific execution parameters and hyperparameters for each component are clearly outlined in their corresponding directories.

4.2 Performance Analysis of Individual and Fused Retrieval Models

In this section, we analyze the impact of various retrieval models and the fused variant on performance across two evaluation phases. Table 2 presents the best scores achieved in Phase 1 and Phase 2 by each model and our proposed LLM-KnowSimFuser which fuses the former models via similarity metrics, allowing for a comparative assessment of their effectiveness and robustness.

Retrieval Model	Phase 1	Phase 2
SFR-Embedding-Mistral	0.20891	0.18659
GritLM-7B	0.20825	0.18622
Linq-Embed-Mistral	0.21208	0.18925
NV-Embed-v1	0.21088	0.18315
Fine-tuned SFR-Embedding-Mistral	0.23160	0.17968
LLM-KnowSimFuser	**0.24621**	**0.20726**

Table 2: Performance comparison in two evaluation phases.

The comparative analysis highlights that while different models exhibit varying degrees of effectiveness and stability, LLM-KnowSimFuser stands out with the highest scores in both evaluation phases. Its ability to maintain strong performance across diverse conditions makes it a highly effective and reliable retrieval model. This performance is indicative of the successful integration of LLM-enhanced representation similarity fusion, enabling more accurate and consistent academic knowledge retrieval. The inclusion of similarity fusion results further underscores the potential of combining multiple models to achieve superior performance, showcasing the benefits of an ensemble approach in enhancing retrieval tasks.

4.3 Investigation into Configurations of Tags and Instructions

In this study, we evaluate the performance of various model configurations by combining different tags and instructions. Table 5 presents the results, showing how each combination impacted the retrieval performance in Phase 2 of this competition. The detailed examination of Table 5 leads to the following conclusions:

Table 3: Tag formatting details.

```
1. {title}\n{body}
2. <question_title> {title} </question_title>
   \n<question_body> {body} </question_body>
3. {title}. {body}
4. Title: {title}\nContent: {body}
5. <title> {title} </title>\n<content> {body}
   </content>
```

Table 4: Different instruction configurations.

```
1. Given a question including title and body,
   retrieve relevant papers that answer the
   question.
2. Given a question including title and body,
   retrieve the paper's title and abstract
   that answer the question.
3. Given a web search query, retrieve relevant
   passages that answer the query.
4. Given a question, retrieve passages that
   answer the question.
```

4.3.1 Tag Effectiveness.

- **Tag 1 (*title\nbody*)**: which demonstrates robust performance across multiple instructions, particularly with the SFR-Embedding-Mistral model, achieving the highest score of 0.18659 with Instruction 2. This suggests that this tag format is well-suited for models that process structured text effectively.
- **Tag 2 ($< question\_title > title < /question\_title > \backslash n < question\_body > body < /question\_body >$)**: which exhibits strong results with the GritLM-7B model. The structured XML format appears to enhance the model's ability to parse and retrieve relevant information, as evidenced by the score of 0.18622 with Instruction 1.
- **Tag 4 ($Title : title\backslash nContent : body$)**: which is the most versatile one, especially with the Linq-Embed-Mistral model. The highest overall performance score of 0.18925 was recorded with Tag 4 and Instruction 2, indicating that this tag format's clear separation of title and content is highly effective for this model.

4.3.2 Instruction Impact.

- **Instruction 2 (Given a question including title and body, retrieve the paper's title and abstract that answer the question.)**: which generally provides the best results across different tags and models. This instruction seems to align well with the models' retrieval mechanisms, suggesting that a focused retrieval objective (title and abstract) enhances performance.
- **Instruction 1 (Given a question including title and body, retrieve relevant papers that answer the question.)**: which also performs well, particularly with Tag 1 and the SFR-Embedding-Mistral model. This indicates that a broader retrieval scope (entire papers) can be effective when paired with suitable tag formats.

[3] https://github.com/loveisp/KDD_2024_AQA

Retrieval Model	Tag	Instruction	Score
SFR-Embedding-Mistral	1	1	0.18390
SFR-Embedding-Mistral	1	2	0.18659
SFR-Embedding-Mistral	1	5	0.18503
GritLM-7B	2	1	0.18622
GritLM-7B	2	2	0.18367
GritLM-7B	2	4	0.18603
Linq-Embed-Mistral	4	1	0.18521
Linq-Embed-Mistral	4	2	0.18925
Linq-Embed-Mistral	4	3	0.18468
Linq-Embed-Mistral	4	4	0.18530
NV-Embed-v1	1	1	0.18103
NV-Embed-v1	3	1	0.18315
NV-Embed-v1	4	1	0.18285
NV-Embed-v1	4	2	0.18251
NV-Embed-v1	4	3	0.18185
NV-Embed-v1	4	4	0.18228
NV-Embed-v1	4	5	0.18174

Table 5: Performance across diverse configurations of tags and instructions.

4.3.3 Model Specific Observations.

- **SFR-Embedding-Mistral**: which consistently performs well with Tag 1 and different instructions, indicating its robustness and adaptability to this tag format.
- **GritLM-7B**: which shows strong performance with Tag 2, highlighting its preference for well-structured tags. The model also performs well with Tag 4 and Instruction 4, suggesting a degree of flexibility in handling structured queries.
- **Linq-Embed-Mistral**: which achieves the highest score overall, particularly with Tag 4 and Instruction 2. This combination's effectiveness underscores the importance of choosing the right tag-instruction pairing for maximizing model performance.
- **NV-Embed-v1**: which shows consistent performance, however it does not achieve the highest scores compared to the other models. The highest score for NV-Embed-v1 was 0.18315 with Tag 3 and Instruction 1, indicating potential areas for optimization.

4.3.4 Possible Directions for Future Work.

- **Further Exploration of Untested Configurations**: there remains potential in exploring the full range of untested tag and instruction combinations. By systematically testing these configurations, it may be possible to discover even more effective pairings that are not covered in this study.
- **Automated Prompt Generation**: developing automated systems to generate and test prompts dynamically could significantly enhance the efficiency of identifying optimal configurations. This approach would allow for a broader exploration of the parameter space and potentially uncover novel configurations that yield superior performance.
- **Focused Optimization for Lower-Performing Models**: specific attention should be directed towards optimizing tags and instructions for models such as NV-Embed-v1. By understanding and addressing the limitations that led to lower performance, it may be possible to enhance the retrieval effectiveness of these models.
- **Model-Specific Tailoring**: customizing tags and instructions based on the characteristics and strengths of individual models could further improve performance. For instance, models that excel with structured tags (like GritLM-7B) could benefit from even more refined tagging strategies.

In conclusion, the findings of this study underscore the significant impact of tag and instruction configurations on retrieval model performance, and strategic selection and optimization of these elements can lead to substantial improvements, and future research should continue to explore and refine these configurations to fully realize their potential.

5 CONCLUSION

In this paper, we presents our solution for the KDD Cup 2024 OAG-Challenge AQA. To tackle this task, we employ a multi-step approach. Firstly, we fine-tune and perform inference using LLM-enhanced pre-trained retrieval models, capitalizing on the powerful language understanding and retrieval capabilities of large language models. Next, we conduct a weighted fusion of the inference results, leveraging a similarity matrix derived from these results to optimize the retrieval performance. Through this meticulous process, our team, *Robo Space*, achieves a commendable final score of 0.20726, and ranks the 2nd place on the final leaderboard.

REFERENCES

[1] Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2022. Tevatron: An Efficient and Flexible Toolkit for Dense Retrieval. *arXiv preprint arXiv:2203.05765* (2022).

[2] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *ICLR*.

[3] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B. *arXiv preprint arXiv:2310.06825* (2023).

[4] Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2021. Billion-Scale Similarity Search with GPUs. *IEEE Trans. Big Data* 7, 3 (2021), 535–547.

[5] Jihoon Kwon Sangmo Gu Yejin Kim Minkyung Cho Jy-yong Sohn Chanyeol Choi Junseong Kim, Seolhwa Lee. 2024. Linq-Embed-Mistral:Elevating Text Retrieval with Improved GPT Data Through Task-Specific Control and Quality Refinement. Linq AI Research Blog. https://getlinq.com/blog/linq-embed-mistral/

[6] Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2024. NV-Embed: Improved Techniques for Training LLMs as Generalist Embedding Models. *arXiv preprint arXiv:2405.17428* (2024).

[7] Niklas Muennighoff, Hongjin Su, Liang Wang, Nan Yang, Furu Wei, Tao Yu, Amanpreet Singh, and Douwe Kiela. 2024. Generative Representational Instruction Tuning. *arXiv preprint arXiv:2402.09906* (2024).

[8] OpenAI. 2023. Chatgpt: Optimizing language models for dialogue.

[9] OpenAI. 2023. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774* (2023).

[10] Fanjin Zhang, Shijie Shi, Yifan Zhu, Bo Chen, Yukuo Cen, Jifan Yu, Yelin Chen, Lulu Wang, Qingfei Zhao, Yuqing Cheng, Tianyi Han, Yuwei An, Dan Zhang, Weng Lam Tam, Kun Cao, Yunhe Pang, Xinyu Guan, Huihui Yuan, Jian Song, Xiaoyan Li, Yuxiao Dong, and Jie Tang. 2024. OAG-Bench: A Human-Curated Benchmark for Academic Graph Mining. *arXiv preprint arXiv:2402.15810* (2024).