Dense Retrieval for Efficient Paper Retrieval in Academic Question Answering

Xuantao Lu*
Xiaohongshu Inc.
Shanghai, China
luxuantao@xiaohongshu.com

Xingwu Hu
China Telecom
Shanghai, China
huxingwu@gmail.com

Abstract

The overarching goal of academic data mining is to deepen our comprehension of the development, nature, and trends of science. It offers the potential to unlock enormous scientific, technological, and educational value. To facilitate related research, Tsinghua University and Zhipu AI have presented the Open Academic Graph Challenge (OAG-Challenge) and published several realistic and challenging datasets [29].

In this paper, we present our solution for the KDD Cup 2024 Academic Question Answering (AQA) task. Participants are required to retrieve the most relevant papers to answer given professional questions from a pool of candidate papers. To address this challenge, we constructed a bi-encoder model for academic paper retrieval. We conducted extensive experiments, exploring various language models (LMs) and ensembling them to boost performance. Additionally, we explored the incorporation of hard negative examples and a reranking model. Our team achieved high-quality results and demonstrated competitive performance in the competition, with mean average precision (MAP) scores of 0.20900 (top-6) and 0.18466 (top-7) on the validation and test sets, respectively. We have released our source code[1].

CCS Concepts

• Information systems → Information retrieval; **• Computing methodologies → Natural language processing**.

Keywords

Information Retrieval, Academic Question Answering, Open Academic Graph Challenge

ACM Reference Format:
Xuantao Lu and Xingwu Hu. 2024. Dense Retrieval for Efficient Paper Retrieval in Academic Question Answering. In *Proceedings of KDD Cup 2024 Workshop: OAG-Challenge (KDDCup '24).* ACM, New York, NY, USA, 4 pages. https://doi.org/XXXXXXX.XXXXXXX

*Corresponding Author
[1] https://github.com/anaivebird/KDD_AQA_2024

KDDCup '24, August 25–29, 2024, Barcelona, Spain

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

Table 1: Statistics of the datasets

	#Size	#Avg. question words	#Avg. body words
Training	8,757	9.25	176.31
Validation	2,919	9.91	148.18
Test	3,000	11.56	103.19
	#Size	#Avg. title words	#Avg. abstract words
Papers	352,651	10.43	159.92

1 Introduction

KDD Cup 2024 OAG-Challenge consists of three tasks: author name disambiguation (AND), academic question answering (AQA), and paper source tracing (PST). In this paper, we focus on the AQA task, which involves retrieving the most relevant papers to answer given professional questions from a pool of candidate papers. In this section, we formalize the dataset and task description.

1.1 Dataset Description

The training set consists of 8757 samples, where each sample includes three fields: question, body, and pids. In this context, body refers to the detailed analysis of the question, and pids represents the paper IDs which are relevant to the question (i.e., positive samples). The validation set and test set have a similar format to the training set but do not include pids. The validation set contains 2919 samples, while the test set contains 3000 samples. Additionally, there are 352,651 candidate papers, each containing pid, title and abstract fields. The statistics of the datasets are summarized in Table 1.

1.2 Task Description

Based on the rich landscape of academic data mining, the AQA task aims to retrieve the most relevant papers to answer given professional questions from a pool of candidate papers. This task plays a crucial role in advancing knowledge acquisition and understanding cognitive impacts within academic research domains. Participants are required to submit a sorted list of the top 20 papers for each question in the test set, and the online evaluation metric used is the top-k mean average precision (MAP) as follows:

$$AP\left(V_q\right) = \frac{1}{R_q} \sum_{k=1}^{M} P_q(k) 1_k \tag{1}$$

$$MAP = \frac{1}{n} \sum_{q=1}^{n} AP\left(V_q\right) \tag{2}$$

where R_q is the number of paper IDs labeled as positives, M represents the total number of candidate papers in the database, n represents the number of samples in the test set, $P_q(k)$ is the precision at the cut-off k in the ranking list of question V_q, and 1_k is an indicator function; $1_k = 1$ if the paper ranked at position k is the correct answer, otherwise $1_k = 0$.

2 METHODOLOGY

The pipeline of our solution is shown in Figure 1 and includes data preprocessing, representation learning, and reranking. We first present the details of data analysis and preprocessing in § 2.1. Representation learning is introduced in § 2.2, and reranking is detailed in § 2.3.

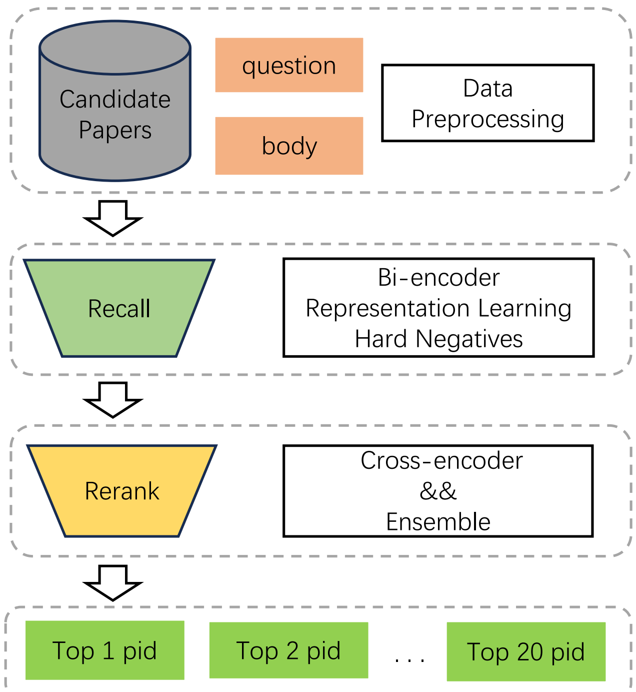

Figure 1: The pipeline of our solution.

2.1 Data Analysis and Preprocessing

We first analyzed the token distribution for the question, body, title, and abstract in the training set. In the experiment, there is not a significant difference in the number of tokens across all models. Here, we take the tokenizer of "Alibaba-NLP/gte-large-en-v1.5" [17] as an example. The results are displayed in Figure 2. It can be observed that the body contains more tokens compared to the other three fields. Additionally, we noted the presence of numerous HTML tags in both the body and abstract, which do not provide useful information. Due to the model's input length limitations, these HTML tags reduce the amount of meaningful information accessible to the model. Therefore, we applied regular expressions to remove HTML tags from the body and abstract fields before training the models.

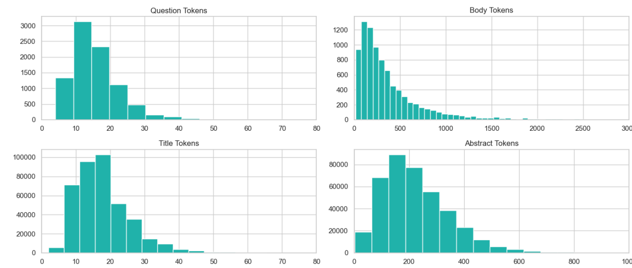

Figure 2: Statistics of tokens in the training set.

2.2 Representation Learning

During the recall stage, an effective method to retrieve relevant papers given a question and body is as follows: Generate embeddings for the question and body, denoted as Q, and generate embeddings for the title and abstract of paper i , denoted as D_i. Sort papers based on the cosine similarity between Q and D_i in descending order:

$$\text{sim}(Q, D_i) = \frac{Q \cdot D_i}{\|Q\| \cdot \|D_i\|} \tag{3}$$

Some efficient similarity search methods and libraries are available here as well, such as HNSW [19] and Faiss [7], but for the size of this dataset, there may not be a significant efficiency improvement.

We use pre-trained language models (PLMs) to construct a bi-encoder framework for representation learning as shown in Figure 3. Text a and text b represent the concatenations of the question with the body, and the title with the abstract, respectively. This approach allows us to encode all papers just once during prediction, which effectively reduces prediction cost.

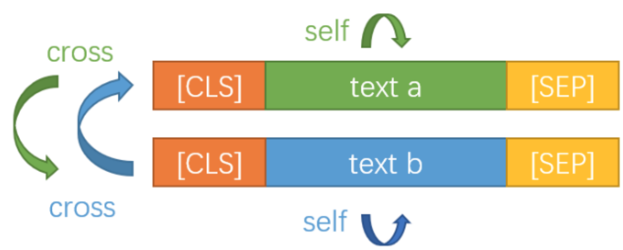

Figure 3: Bi-encoder framework.

We model this task as a binary classification problem and train the model using binary cross-entropy (BCE) loss. Since the dataset only provides positive papers corresponding to query and body, we had to construct negatives. We tried three different approaches:

- For each sample, randomly select N papers from the paper pool as negatives.
- Use unfine-tuned pre-trained language models to calculate $sim(Q, D_i)$ and select the top N papers with the highest scores (excluding positive samples) as hard negatives.
- For each sample, select N papers from the paper pool as negatives, ensuring that after selecting negatives for all samples, each paper in the pool is selected at least once. This ensures that representations of all candidate papers are trained.

The experimental results indicate that the third method achieved the best performance. We will provide a detailed discussion of this in § 3.3.

2.3 Reranking

Generally, it is crucial to rerank the recalled results to achieve better performance. We explored two different reranking approaches. One involved constructing a cross-encoder model, as shown in Figure 4, which leveraged cross-attention between the question, body, and paper to capture more semantic information. The other approach entailed training multiple bi-encoder models and ensembling the prediction results.

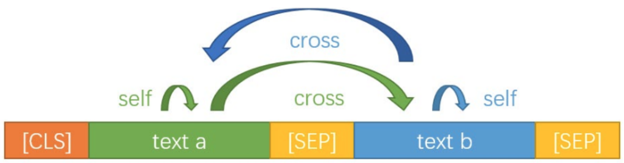

Figure 4: Cross-encoder framework.

We selected the top 200 papers with the highest scores from the recall results for reranking. For the cross-encoder, we directly use its prediction scores as the final scores. For the model ensemble, we aggregate the prediction scores from multiple bi-encoder models. We then select the top 20 papers with the highest scores as the final results.

3 EXPERIMENTS

In this section, we present our main experiment results and discuss the findings.

3.1 Experimental Setup

For the bi-encoder, we experimented with different PLMs, including snowflake-arctic-embed-l [21], gte-large-en-v1.5 [17], bge-large-en-v1.5 [26], mxbai-embed-large-v1 [15] and UAE-Large-V1 [16]. For the cross-encoder, we used bge-reranker-large [26]. The experiments were conducted on the Linux operating system, utilizing PyTorch[2], transformers [25], and FlagEmbedding [3] for implementation.

3.2 Overall Performance

We compare the performance of different methods on the validation and test sets in Table 2. The default is to use the third method of negatives selection outlined in § 2.2. From the results, we conclude that:

1) When using a single model, the gte model achieved the best performance and demonstrated significant advantages compared to other models.

2) When using model ensemble, the combination of gte and snowflakes achieved the best performance. Additionally, it can be observed that more models in the ensemble do not necessarily lead to better performance. Ensembling a poor performing model could potentially lead to a decline in overall performance.

[2]https://pytorch.org

Table 2: Overall Performance (MAP@20)

Method	Validation	Test
bge	0.1804	0.1529
UAE	-	0.1546
mxbai	-	0.1554
snowflake	-	0.1607
gte	0.1938	0.1724
gte+reranker	0.1903	-
bge+gte+mxbai	0.1881	0.1762
bge+gte+mxbai+snowflakes+UAE	0.1934	0.1797
bge+gte+mxbai+snowflakes	0.2023	0.1830
gte+snowflakes	**0.2090**	**0.1846**

Table 3: The performance (MAP@20) under different negatives selection strategies

Method		Validation
bge	w/ hard negatives	0.1053
	w/ random negatives	0.1618
	w/ negatives covering all papers	**0.1804**

3) The introduction of a reranker did not lead to performance improvement. We feel there might be room for improvement in the construction of training data or training methods for the reranker.

3.3 Negatives Selection

The performance under different negatives selection strategies is shown in Table 3. It can be observed that the introduction of hard negatives significantly deteriorated the model's performance. These hard negatives were considered by unfine-tuned PLMs to have higher similarity with the question and body. We found that some of these papers were actually positives that had been mislabeled. Treating these samples as negatives led to a decline in the model's performance. Furthermore, ensuring that each paper in the pool is selected at least once tends to result in better performance compared to randomly selecting negatives.

4 RELATED WORK

In question answering (QA), the passage retriever is crucial for identifying relevant passages for extracting answers. Traditional methods, such as TF-IDF and BM25, have used term-based retrievers but are limited in their representation capabilities [2]. Recent advancements have leveraged deep learning to enhance these retrievers, incorporating techniques like document expansion [23], question expansion [20], and term weight estimation [4].

Unlike these term-based methods, dense passage retrieval has emerged, representing both questions and documents as dense vectors (i.e., embeddings) within a bi-encoder framework. Current approaches fall into two categories: self-supervised pre-training for retrieval [1, 9, 14] and fine-tuning pre-trained language models on labeled datasets. Although the bi-encoder architecture is promising, training such a retriever effectively is challenging. It faces issues like training and inference discrepancies, a large number of unlabeled positives, and limited training data. Recent studies[1, 10, 13, 18] have attempted to address the first issue by developing complex sampling mechanisms to create hard negatives but still struggle with false negatives. The other two challenges have been less frequently tackled in QA.

The concept of using dense vector representations in retrieval is not new, with roots in Latent Semantic Analysis [6]. Recently, discriminatively trained dense encoders with labeled query-document pairs have gained traction [8, 11, 28], applied in areas like cross-lingual document retrieval, ad relevance prediction, web search, and entity retrieval. These dense methods complement sparse vectors by scoring semantically related text pairs highly, even without exact word overlap. However, dense representations often underperform compared to sparse models.

Although not the primary focus here, dense representations from pre-trained models combined with cross-attention mechanisms have shown potential in re-ranking passages or dialogues [12, 22]. In QA, Das et al. [5] introduced an iterative retrieval approach using reformulated question vectors, while Seo et al. [24] proposed bypassing passage retrieval altogether by directly encoding answer phrases as vectors for retrieval. Lee et al. [14] jointly trained question encoders and readers with additional pre-training to align question surrogates with relevant passages, outperforming BM25 plus reader methods in QA accuracy. REALM [9] furthered this by asynchronously tuning the passage encoder during training via re-indexing. Improvements in pre-training objectives have also been seen with work by Xiong et al. [27].

5 CONCLUSION

In this paper, we introduce our pipeline for the KDD CUP 2024 OAG-Challenge AQA task. We constructed a bi-encoder model for academic paper retrieval, experimented with different LMs, and ensembled them to boost performance. Furthermore, we tried different ways of constructing negative samples and introduced a rerank model. Our team achieved high-quality results and demonstrated competitive performance in the competition.

Acknowledgments

This work received support from Xiaohongshu Inc. and China Telecom, for which we are sincerely grateful.

References

[1] Wei-Cheng Chang, Felix X. Yu, Yin-Wen Chang, Yiming Yang, and Sanjiv Kumar. 2020. Pre-training Tasks for Embedding-based Large-scale Retrieval. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020.*

[2] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to Answer Open-Domain Questions. 1870–1879.

[3] Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2023. BGE M3-Embedding: Multi-Lingual, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation. arXiv:2309.07597 [cs.CL]

[4] Zhuyun Dai and Jamie Callan. 2019. Deeper Text Understanding for IR with Contextual Neural Language Modeling. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*, Benjamin Piwowarski, Max Chevalier, Éric Gaussier, Yoelle Maarek, Jian-Yun Nie, and Falk Scholer (Eds.). 985–988.

[5] Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, and Andrew McCallum. 2019. Multi-step retriever-reader interaction for scalable open-domain question answering.

[6] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science* 41, 6 (1990), 391–407.

[7] Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The Faiss library. (2024). arXiv:2401.08281 [cs.LG]

[8] Daniel Gillick, Sayali Kulkarni, Larry Lansing, Alessandro Presta, Jason Baldridge, Eugene Ie, and Diego Garcia-Olano. 2019. Learning Dense Representations for Entity Retrieval.

[9] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. REALM: Retrieval-augmented language model pre-training. *ArXiv* abs/2002.08909 (2020).

[10] Matthew L. Henderson, Rami Al-Rfou, Brian Strope, Yun-Hsuan Sung, László Lukács, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. 2017. Efficient Natural Language Response Suggestion for Smart Reply. *CoRR* abs/1705.00652 (2017).

[11] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for Web search using clickthrough data. 2333–2338.

[12] Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2020. Poly-encoders: Architectures and Pre-training Strategies for Fast and Accurate Multi-sentence Scoring.

[13] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick S. H. Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (Eds.). 6769–6781.

[14] Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent Retrieval for Weakly Supervised Open Domain Question Answering. 6086–6096.

[15] Sean Lee, Aamir Shakir, Darius Koenig, and Julius Lipp. 2024. *Open Source Strikes Bread - New Fluffy Embeddings Model.* https://www.mixedbread.ai/blog/mxbai-embed-large-v1

[16] Xianming Li and Jing Li. 2023. AnglE-optimized Text Embeddings. *arXiv preprint arXiv:2309.12871* (2023).

[17] Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281* (2023).

[18] Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2020. Sparse, Dense, and Attentional Representations for Text Retrieval. *CoRR* abs/2005.00181 (2020).

[19] Yu A Malkov and Dmitry A Yashunin. 2018. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE transactions on pattern analysis and machine intelligence* 42, 4 (2018), 824–836.

[20] Yuning Mao, Pengcheng He, Xiaodong Liu, Yelong Shen, Jianfeng Gao, Jiawei Han, and Weizhu Chen. 2020. Generation-Augmented Retrieval for Open-domain Question Answering. *CoRR* abs/2009.08553 (2020).

[21] Luke Merrick, Danmei Xu, Gaurav Nuti, and Daniel Campos. 2024. Arctic-Embed: Scalable, Efficient, and Accurate Text Embedding Models. *arXiv preprint arXiv:2405.05374* (2024).

[22] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *ArXiv* abs/1901.04085 (2019).

[23] Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. 2019. Document Expansion by Query Prediction. *CoRR* abs/1904.08375 (2019).

[24] Minjoon Seo, Jinhyuk Lee, Tom Kwiatkowski, Ankur Parikh, Ali Farhadi, and Hannaneh Hajishirzi. 2019. Real-Time Open-Domain Question Answering with Dense-Sparse Phrase Index.

[25] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, Online, 38–45. https://www.aclweb.org/anthology/2020.emnlp-demos.6

[26] Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-Pack: Packaged Resources To Advance General Chinese Embedding. arXiv:2309.07597 [cs.CL]

[27] Wenhan Xiong, Hankang Wang, and William Yang Wang. 2020. Progressively Pretrained Dense Corpus Index for Open-Domain Question Answering. *ArXiv* abs/2005.00038 (2020).

[28] Wen-tau Yih, Kristina Toutanova, John C Platt, and Christopher Meek. 2011. Learning discriminative projections for text similarity measures. 247–256.

[29] 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).