A Multi-Channel Retriever for Effective Academic Question Answering

Fuxin Jiang
Academy of Mathematics and Systems
Science, Chinese Academy of Sciences
Beijing, China
jiangfuxin17@mails.ucas.ac.cn

Quanying Lu[*]
College of of Economics and Management,
Beijing University Of Technology
Beijing, China
luquanying@bjut.edu.cn

Abstract

In this era of booming technology and rapidly updated information, it has become a top priority to provide researchers and the general public with high-quality cutting-edge academic knowledge in multiple fields. Accurate academic paper retrieval can help researchers quickly capture the frontiers of their fields and accelerate research progress. For this purpose, we propose a multi-channel retriever that includes a Naïve Embedding-based Retriever and a Graph Embedding-based Retriever that considers the citation relations between papers. We then use Reciprocal Rank Fusion (RRF) to ensemble the results from the multiple retrievals. Our approach achieved fifth-place position in the KDD Cup 2024 AQA Challenge. Code is publicly available at https://github.com/fuxinjiang/KDDCUP-2024_AQA.

CCS Concepts

• **Information systems** → *Information retrieval.*

Keywords

Multiple-channel Retriever, Ensemble

ACM Reference Format:
Fuxin Jiang and Quanying Lu. 2024. A Multi-Channel Retriever for Effective Academic Question Answering. In *Proceedings of KDD 2024 Workshop OAG-Challenge Cup (KDDCup'24) (KDDCup '24).* ACM, New York, NY, USA, 4 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Effective academic question answering (AQA) plays an important role in academic research and has a significant impact on the effectiveness and comprehensiveness of research results. The accuracy of literature retrieval ensures that researchers obtain relevant and high-quality information, thereby improving the integrity of their work. In the context of the ever-expanding academic literature, the ability to formulate precise queries and use advanced retrieval techniques is essential [22].

To further enhance the efficacy of academic question answering, we propose a multi-channel retriever that includes a Naïve Embedding-based Retriever and a Graph Embedding-based Retriever that considers the citation relationships between papers. We then use reciprocal rank fusion (RRF) to ensemble the results from the multiple retrievals. Although the proposed retrieval method is relatively simple, it achieved satisfactory results on real-world data, securing the fifth-place position in the KDD Cup 2024 AQA Challenge.

2 Relted Works

Text retrieval involves the task of generating a ranked list of documents in response to a given query. Approaches to text retrieval are commonly categorized into two types: lexical methods, which are based on word matching [7, 16], and semantic methods, which aim to assess document relevance by interpreting the underlying meaning beyond the surface level of words [3, 19]. While traditional lexical retrieval algorithms, which have been established for decades, continue to perform effectively, it is the machine learning techniques that have shown the greatest promise in enhancing semantic retrieval capabilities [2, 4, 20].

Embedding of text [4, 5] encodes its semantic information and can be employed in many downstream applications, such as retrieval, reranking, classification, clustering, and semantic textual similarity tasks. The embedding-based retriever is also a critical component for retrieval-augmented generation (RAG) [10], which allows large language models (LLMs) to access the most up-to-date external or proprietary knowledge without modifying the model parameters [13, 18]. Currently, embedding methods mainly include bidirectional embedding models and decoder-only LLM-based embedding models [9]. Bidirectional embedding models mainly fine-tune parameters based on BERT [5] and T5 [14]. For example, SentenceBERT [15] and SimCSE [6], which fine-tune BERT on natural language inference (NLI) datasets. Decoder-only LLM-based embedding models are mainly fine-tuned based on LLMs. For example, LLM2Vec [1] uses a specially designed masked token prediction to warm-up the bidirectional attention and unsupervised contrastive learning to improve the effect of text embedding. NV-Embed [9] significantly enhances the performance of large language models in general text embedding tasks by introducing a latent attention layer, removing causal attention masks, and employing a two-stage instruction-tuning.

Graph neural networks (GNNs) have also attracted considerable attention in question answering systems. For example, QA-GNN [21] leverages language models and knowledge graphs to enhance interpretability and structured reasoning capabilities, Liu

[*] Corresponding author

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

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

et al. propose a GNN-encoder model in which query (passage) information is fused into passage (query) representations via graph neural networks that are constructed by queries and their top retrieved passages [12].

3 Methodology

We propose a multi-channel retriever based on a naïve embedding retriever and graph embedding retriever that considers node relationships, as shown in Fig. 1.

3.1 Naïve Embedding-based Retriever

Firstly, following the work of Karpukhin et al. [7], we employ a dense encoder $E(\cdot)$ that maps each paper to a d-dimensional real-valued vector and constructs an index for all papers P used for retrieval. We also apply the same encoder $E(\cdot)$ to map the input question to a d-dimensional vector, and retrieve k papers whose vectors are closest to the question vector. Define the similarity between the question q and the paper $p \in P$ using the inner product of their vectors as follows:

$$sim(q, p) = E(q)^T E(p)$$

At the same time, with the continuous optimization of model structures and the increasing volume of pre-training data, the representation of text by pre-trained models has significantly improved. Therefore, we employ the pre-trained models NV-Embed [9] and GTE [11] as dense encoders and fine-tune them separately on the dataset utilized for this academic paper retrieval task. This allows us to achieve more precise representations of texts specific to this particular task.

Training: Fine-tuning the encoders so that the dot-product similarity becomes a good ranking function aims to create a vector space such that relevant pairs of questions and papers will have smaller distances (i.e., higher similarity) than the irrelevant ones, by learning a better embedding function. Let $\mathcal{D} = \{q_i, p_i^+, p_{i,1}^-, p_{i,2}^-, ..., p_{i,n}^-\}_{i=1}^m$, we optimize the loss function as the negative log likelihood of the positive sample:

$$\mathcal{L}(q_i, p_i^+, p_{i,1}^-, p_{i,2}^-, ..., p_{i,n}^-) = -log \frac{e^{sim(q_i,p_i^+)}}{e^{sim(q_i,p_i^+)} + \sum_{j=1}^{n} e^{sim(q_i,p_j^-)}}$$

where p_i^+ represents the correct answer to question q_i (positive sample), and $p_{i,\cdot}^-$ represents the wrong answer to question q_i (negative sample).

Negative Samples: For retrieval tasks, positive samples are usually explicitly available, while negative samples need to be selected from a large pool. For instance, in academic question-answering dataset, papers related to the question are provided. All other papers in the collection, although not explicitly designated, can be implicitly considered irrelevant. In practice, the method of selecting negative samples is often overlooked, yet it can be crucial for training high-quality encoders. We consider two different types of negative samples: (1) Random: any random paper from the dataset; (2) Hard negatives: samples inferred through an initial pre-trained model but not identified as positive samples.

3.2 Graph Embedding-based Retriever

The naïve embedding retriever only considers the semantic similarity between the users' questions and the papers, but it ignores the relationships between papers (e.g., citations). Therefore, we additionally introduce an additional retriever based on Graph Convolutional Network (GCN) [8], which incorporates the relational structure between papers.

In order to avoid over-smoothing, we directly use the first-order graph convolutional operation. The specific process is as follows:

Step 1. We utilize DBLP dataset [17] to construct the text attribute graph $\mathcal{G}(\mathcal{V}, \mathbf{A})$ of papers. Specifically, if a paper cites other papers, then the adjacency matrix entry $\mathbf{A}_{i,j} = 1$; otherwise, $\mathbf{A}_{i,j} = 0$.

Step 2. Subsequently, the encoder $E(\cdot)$ is utilized to embed the user question q and all papers P. Then, graph convolutional operation is employed to aggregate node information. The message passing is defined as

$$H(P) = (\tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-1/2})E(P)\mathbf{W}_P + \mathbf{b}_P$$

where $\tilde{\mathbf{D}} = \mathbf{D} + \mathbf{I}$ and $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ represent the degree and adjacency matrices enhanced with self-loops, D is the diagonal degree matrix, $\mathbf{W}_P$ and $\mathbf{b}_P$ are learnable parameters.

Step 3. To achieve an equitable enhancement of user question representation, we introduce the learnable parameters $\mathbf{W}_Q$ and $\mathbf{b}_Q$ as:

$$EE(q) = E(q)\mathbf{W}_Q + \mathbf{b}_Q$$

Step 4. The training process, similar to that of the naïve embedding retriever. The similarity between the question and the paper using the inner product of their vectors as:

$$sim(q, p) = EE(q)^T H_p$$

where H_p is the representation of $H(P)$ to paper p.

3.3 Model Ensemble

Firstly, we select the 100 papers P_{cand}^{NE} most similar to the user question from the naïve embedding-based retriever (NE) through similarity matching, and also select the 100 papers P_{cand}^{GE} most similar to the user question from the graph embedding-based retriever (GE).

Then we use Reciprocal Rank Fusion (RRF) to ensemble the retrieval results of navë embedding-based retriever and graph embedding-based retriever. For each paper p in the candidate papers $\{P_{cand}^{NE}, P_{cand}^{GE}\}$ generated by each retrieval method, we compute:

$$RRF_{score}(p) = \sum_{r\in\{NE,GE\}} \frac{1}{k + r(p)}$$

where $r(p)$ represents the ranking of paper p in retriever r. The constant k is set to 60 for this task.

Finally, we select the 20 papers with the largest RRF_{score} as the final retrieval results.

4 Experiments

4.1 Datasets

We use the OAG-QA [22] dataset to verify the effectiveness of our proposed method, which aggregates question-paper pairs from

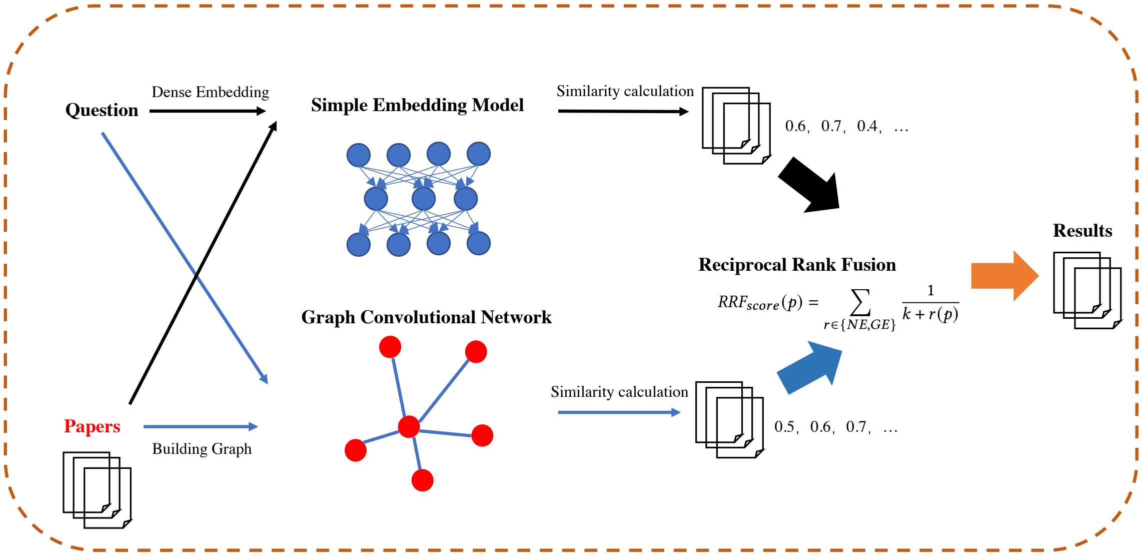

Figure 1: The framework of th proposed multi-channel retriever

academic question-answering platforms such as StackExchange and Zhihu. It mainly includes training data, validation data, test data and candidate paper list. The amount of data contained in each part is shown in the Table 1.

Table 1: Statistics of the OAG-QA dataset

Data	Amount	Key words
Train	8757	"question", "body", "pids"
Valid	2919	"question", "body"
Test	3000	"question", "body"
Papers List	466387	"pid", "title", "abstract"

The training dataset includes three key elements: "question", "body", and "pids", These elements represent a concise description of the question, a detailed exposition of the question, and a list of paper identifiers (pids) associated with relevant papers, respectively. The validation dataset and test dataset do not have "pids". Conversely, in the candidate papers list, the papers are paired with their content, including titles and abstracts, through a dictionary that maps the paper identifier (pid).

4.2 Experimental Settings

We selected the pre-trained models NV-Embed[9] and GTE[11] as dense encoders. The training parameters are set as follows: batch size of 24, learning rate of 1e-5, and 5 epochs. We perform the experiments on a Debian server with Intel(R) Xeon(R) Platinum 8336C CPU @ 2.30GHz, 920GB RAM and 8 A100 80G GPUs.

4.3 Results

Metrics: We utilize Mean Average Precision (MAP) and top-k MAP as evaluation metrics. For each query V_q, the Average Precision (AP) is calculated using the following formula:

$$AP(V_q) = \frac{1}{R_q} \sum_{k=1}^{M} P_q(k) \cdot \frac{1}{k}$$

where R_q is the number of papers marked as positive examples, M is the number of documents in the database, and $P_q(k)$ is the precision at rank k in the ranking list for query V_q. The term $\frac{1}{k}$ is an indicator function that equals 1 if the k-th returned document is relevant, and 0 otherwise.

For a given set of n queries, the MAP is calculated as follows:

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

The top-K MAP can be similarly computed by setting $M = K$ in the above equation.

Model Comparison: Based on the result presented in Table 2, we conduct a comparative analysis of retrieval accuracy across different models on a test set. The GTE model achieved a Mean Average Precision (MAP) score of 0.17747, which positioned it at rank 10. In contrast, the NV-Embed model obtained a MAP score of 0.18683, securing the 5th rank. Similarly, the Ensemble model slightly outperformed the NV-Embed with a MAP score of 0.18688, also ranking 5th.

These results indicate that both the NV-Embed and Ensemble model demonstrate superior retrieval accuracy compared to the GTE model. Notably, while the Ensemble model has a marginally higher MAP score than the NV-Embed model, both models are ranked equally. This suggests that the Ensemble model's incremental improvement in MAP may not significantly impact its relative ranking in this context.

5 Conclusion

We propose a multi-channel retriever that includes a Naïve Embedding-based Retriever and a Graph Embedding-based Retriever that considers the citation relations between papers. We then use Reciprocal

Table 2: Comparison of retrieval accuracy on test set

Model	MAP	Rank
GTE	0.17747	10
NV-Embed	0.18683	5
Ensemble	0.18688	5

Rank Fusion (RRF) to ensemble the results from the multiple retrievals. Our approach achieved fifth-place position in the KDD Cup 2024 AQA Challenge.

References

[1] Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. Llm2vec: Large language models are secretly powerful text encoders. *arXiv preprint arXiv:2404.05961* (2024).

[2] ZeFeng Cai, Chongyang Tao, Tao Shen, Can Xu, Xiubo Geng, Xin Alex Lin, Liang He, and Daxin Jiang. 2022. HypeR: Multitask Hyper-Prompted Training Enables Large-Scale Retrieval Generalization. In *The Eleventh International Conference on Learning Representations*.

[3] 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. *arXiv preprint arXiv:2402.03216* (2024).

[4] Zhuyun Dai, Vincent Y Zhao, Ji Ma, Yi Luan, Jianmo Ni, Jing Lu, Anton Bakalov, Kelvin Guu, Keith B Hall, and Ming-Wei Chang. 2022. Promptagator: Few-shot dense retrieval from 8 examples. *arXiv preprint arXiv:2209.11755* (2022).

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[6] Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821* (2021).

[7] 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* (2020).

[8] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[9] 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).

[10] 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 (2020), 9459–9474.

[11] 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).

[12] Jiduan Liu, Jiahao Liu, Yang Yang, Jingang Wang, Wei Wu, Dongyan Zhao, and Rui Yan. 2022. GNN-encoder: Learning a Dual-encoder Architecture via Graph Neural Networks for Dense Passage Retrieval. In *Findings of the Association for Computational Linguistics: EMNLP 2022*. 564–575.

[13] Zihan Liu, Wei Ping, Rajarshi Roy, Peng Xu, Chankyu Lee, Mohammad Shoeybi, and Bryan Catanzaro. 2024. Chatqa: Surpassing gpt-4 on conversational qa and rag. *arXiv preprint arXiv:2401.10225* (2024).

[14] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research* 21, 140 (2020), 1–67.

[15] Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084* (2019).

[16] Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval* 3, 4 (2009), 333–389.

[17] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. 2008. ArnetMiner: Extraction and Mining of Academic Social Networks. In *KDD'08*. 990–998.

[18] Boxin Wang, Wei Ping, Lawrence McAfee, Peng Xu, Bo Li, Mohammad Shoeybi, and Bryan Catanzaro. 2023. Instructretro: Instruction tuning post retrieval-augmented pretraining. *arXiv preprint arXiv:2310.07713* (2023).

[19] Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533* (2022).

[20] Canwen Xu, Daya Guo, Nan Duan, and Julian McAuley. 2022. Laprador: Unsupervised pretrained dense retriever for zero-shot text retrieval. *arXiv preprint arXiv:2203.06169* (2022).

[21] Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. 2021. QA-GNN: Reasoning with language models and knowledge graphs for question answering. *arXiv preprint arXiv:2104.06378* (2021).

[22] 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. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.