3rd Place Solution to KDD Cup 2024 Task 2: Large Academic Graph Retrieval via Multi-Track Message Passing Neural Network

Yu Li*
Taile Chen*
Jingxin Hai*
Hongbin Pei
Jing Tao†
MOE KLINNS Lab
Xi'an, China
{liyu1998,peihongbin}@xjtu.edu.cn

Yixuan Zhou
Xi'an Jiaotong University
Xi'an, China
yxzhou@stu.xjtu.edu.cn

Yong Fu
Tieying Zhang
Bytedance.Inc
San Jose, USA
tieying.zhang@bytedance.com

Abstract

In the realm of academic research, accessing precise and useful information is paramount for scholars. However, traditional academic question answering (AQA) systems based on the dual-tower model's often falter when dealing with complex citation networks. To overcome these challenges, we introduce the "Multi-Track Textual Graph Retriever," an innovative system that harnesses the power of a Paper Classifier, Multi-Track Textual Graph Retriever, and dynamic Retrieval Task Training. Our approach not only integrates text features and citation relationships into a cohesive framework but also employs a dynamic sampling strategy based on hard negative mining. This strategy dynamically refines the training process, leading to significant improvements in the accuracy and efficiency of academic paper retrieval. Finally, we achieved high-quality retrieval result and demonstrated competitive performance in the KDD CUP 2024 OAG-Challenge Task 2 competition (3rd place). The code is available at https://github.com/liyu199809/PineappleHouse.

CCS Concepts

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

Keywords

Academic graph mining, information retrieval.

1 Introduction

Academic question answering (AQA) is of great practical significance as it helps researchers obtain accurate and useful information in the academic field. The task give professional questions and a pool of candidate papers and the objective is to retrieve the most relevant papers to answer these questions. Existing methods are based on the dual-tower model established by pure text semantics and recalled through similarity. In the AQA scenario, the existing methods cannot consider the citation interaction between papers. This leads to a high concentration of recall results then get poor retrieval performance.[3] Simply using GNN for recall would lose a lot of semantic information. Moreover, in the large graph scenario, due to the existence of the over-squashing problem, the receptive field of GNN is only 2-3 hops, which also brings the loss of structural information.

To solve this problem, we design a simple yet efficient solution called **citation-aware academic graph retriever** for modeling the text features and the citation interaction between papers simultaneously. The system is implemented through three primary modules: **(1) paper classifier:** It uses in-context learning with prompts to classify documents based on a few annotated examples. **(2) multi-track textual graph retriever:** Constructs a text-attributed graph to model citation relationships using a multi-track message passing mechanism, preventing oversmoothing and oversquashing in GNNs. **(3) retrieval task training:** Employs InfoNCE loss for training to discern relevant from irrelevant text pairs, with a dynamic sampling approach based on hard negative mining to refine the training set continuously. Despite its simplicity, our approach yields high-quality retrieve result, earning us the third-place position in KDD CUP 2024 Task 2.

2 Related Works

The embedding-based retriever (EBR) implement retrieval based on the semantic information of the text. With the rise of the pretrained language models [4], the accuracy of EBR has been substantially improved, making it a critical recall path in retrieval-augmented generation (RAG) [10], reranking, classification and many other downstream applications. Bidirectional embedding models, such as BERT [5] have long been the dominant approaches for general-purpose embedding tasks. However the latest study by Wang et al. [23] demonstrates that decoder-only LLMs can surpass bidirectional embedding models [1, 15, 22] in retrieval and general-purpose embedding tasks.

The fusion of graph neural networks (GNNs) and LLMs has produced promising outcomes across various domains, such as node categorization [7, 9], graph categorization [17, 26], and utilizing LLMs for tasks related to knowledge graphs [12, 21], etc. He et al. [8] first propose a RAG method for textual graphs to enhance LLM's understanding of graph structure. Some graph-based RAG methods utilize the topology of knowledge graphs or paragraph graphs to enhance the LLM's understanding of text [3, 14], which performs well in large-scale text graphs.

3 Method

Our task is to train a retriever that can precisely retrieve the most relevant papers for any professional academic question. Note that

*Both authors contributed equally to this research.
†Corresponding author

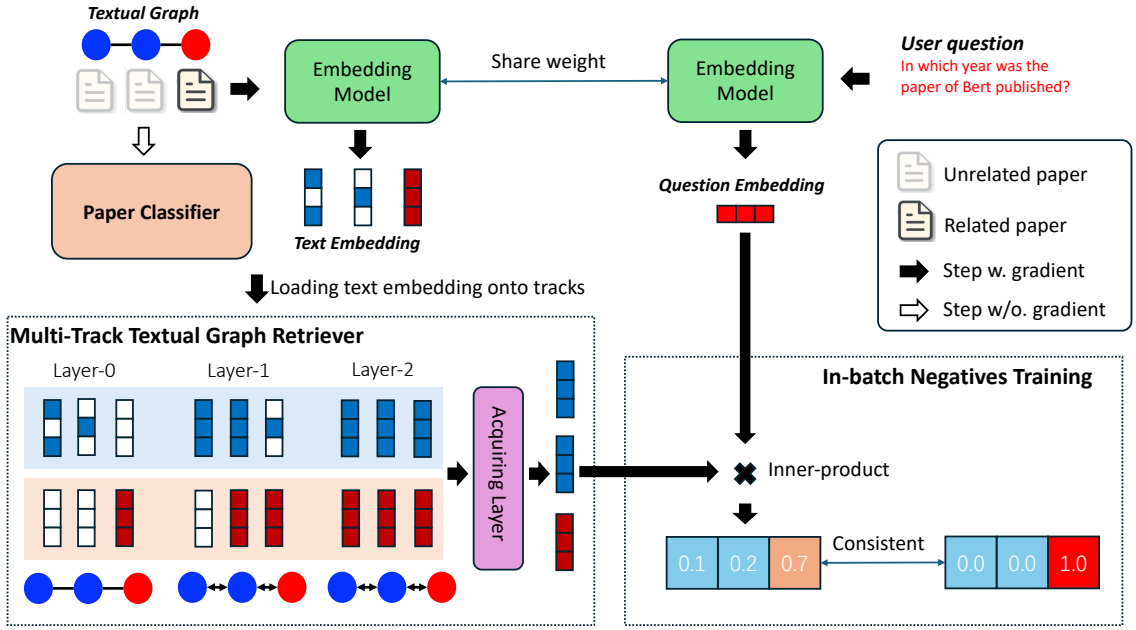

Figure 1: The framework of our method

the retriever is trained by question-paper pairs $\{(q_i, p_i)\}_{i=1}^{n}$. Compared with the basic retrieval task, there are a large number of citation relationships that need to be accurately modeled in the academic question answering task, and academic-related questions are more difficult than task-oriented QA, so a more accurate similarity modeling is required.

To meet these requirements, we design three modules as Fig.1.

3.1 Paper Classifier

Traditional pre-training and fine-tuning approaches[4] require a large amount of annotated data to work well, but in-context learning can generate accurate responses with only a few annotated examples[13]. This is particularly useful when dealing with large datasets that have very few annotations. So, We fellow the prompt-based in-context learning paradigm to build a document classifier.

Given an input sequence $\boldsymbol{x}_{input} = \{x_0, x_1, \cdots, x_L\}$, language model needs to generate a predefined text response $\boldsymbol{y} \in \mathcal{Y}$(e.g., computer science etc.) under the prompt $\boldsymbol{x}_{prompt}$.

$\boldsymbol{x}_{prompt}$ consists of the following three components:

(1) **Task description** $\boldsymbol{x}_{desc}$. Taking text classification task in this paper as an example, the task description is as follows: *You are a professional text classifier. Please classify the given text into the category of Computer Science, $\cdots$,etc.*

(2) **Demonstration** $\mathcal{D}_{demo}$ consists of a series of annotated data. The purpose is to provide helpful examples and output format to the language model. $\boldsymbol{x}_{demo}$ used in this paper are presented as follows:
$\{(\boldsymbol{x}^0_{demo}, \boldsymbol{y}^0_{demo}), (\boldsymbol{x}^1_{demo}, \boldsymbol{y}^1_{demo}), \cdots, (\boldsymbol{x}^k_{demo}, \boldsymbol{y}^k_{demo})\}$
where $\boldsymbol{x}^i_{demo}$ means j-th example query and $\boldsymbol{y}^i_{demo}$ corresponds to the respective example label.

(3) **Test input** $\boldsymbol{x}_{test}$ is the text sequence that we need to classify.

Therefore, the $\boldsymbol{x}_{prompt}$ is as follows:

$$\boldsymbol{x}_{prompt} = \{\boldsymbol{x}_{desc}, \backslash n, \mathcal{D}_{demo}, \backslash n, \boldsymbol{x}_{test}\}$$

$\mathcal{D}_{demo}$ is sampled from the 100 artificially labeled dataset $\mathcal{D}_{train}$. When labeling the data, we should be pay attention to the balance of categories, and the classification standard of paper categories is consistent with [19].

Furthermore, the sampling strategy of $\mathcal{D}_{demo}$ has a significant impact on the final classification performance. Maintaining semantic similarity between $\mathcal{D}_{demo}$ and $\boldsymbol{x}_{test}$ is one of the effective strategies[11]. Therefore, we use a good pre-trained model (such as *bge-large-en-v1.5*) to select the three most relevant labeled data from $\mathcal{D}_{train}$.

Although in-context learning has solved the problem of limited label, its token cost and time cost are very high on extremely large datasets. To solve this problem, we establish a journal-paper mapping by [20]. The specific category of a paper is inferred through the journal in which the paper is included. This method has reduced the number of inferences by five times.

3.2 Multi-Track Textual Graph Retriever

We utilize the additional dataset[20] provided by the official to obtain the citation relationships between papers. Based on this, we construct a text-attributed graph $\mathcal{G} = (\mathcal{V}, \mathbf{A}, \boldsymbol{s}_v)$. Specifically, If a paper i cites others documents j then adjacency matrix $\mathbf{A}_{i,j} = 1$. Otherwise, $\mathbf{A}_{i,j} = 0$. Each node $n \in \mathcal{V}$ is associated with a sequential text feature(sentences) $\boldsymbol{s}_n$.

Motivated by [16], we employ multi-track message passing mechanism to model the citation relationships between papers. Messages passing within *tracks* greatly avoids the issues of oversmoothing and oversquashing, enabling GNNs to embed richer structural information from $\mathcal{G}$. Note that message tracks $\mathcal{T}$ are defined as a set of

isomorphic graphs mirroring the topology of $\mathcal{G}$. Each track $T \in \mathcal{T}$ is uniquely used for passing messages corresponding to a specific node category generate by Section 3.1. Next, we will provide a detailed step-by-step explanation of the retriever.

Step1:Loading. All nodes initial features $\mathbf{X}$, which are generated by text sequence encoder $f_{text}(\cdot)$, are loaded onto the corresponding tracks $\mathcal{T}$ as initial messages $\mathcal{M}^{(0)}$. The sending guidance matrix $\mathbf{F}^s \in \{0,1\}^{|\mathcal{T}|\times|\mathcal{V}|}$ determines which specific track each node will be sent to.

$$\begin{aligned}\mathcal{M}^{(0)}_{T,v,:} &= \mathbf{X}_v \text{ if } \mathbf{F}^s_{T,v} = 1\\ \mathcal{M}^{(0)}_{T,v,:} &= \overrightarrow{\mathbf{0}} \text{ if } \mathbf{F}^s_{T,v} = 0,\end{aligned} \tag{1}$$

Ideally, each paper node should be sent to only one track. However, due to the lack of labeling information and the potential loss in papers-journal mapping process names in Section 3.1, we will send each node to the three most probable tracks to improve track loading recall.

Step2:Multi-Track Message Passing(MTMP). The initial messages are updated by propagating and aggregating in respective tracks over L iterations.

$$\mathcal{M}^{(\ell)}_{T,:,:} = (\tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-1/2})\mathcal{M}^{(\ell-1)}_{T,:,:} + \alpha_\ell \mathcal{M}^{(0)}_{T,:,:}, \tag{2}$$

where messages $\mathcal{M}^{(\ell-1)}_{T,:,:}$ in neighborhood are aggregated according to the normalized adjacency matrix $\tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-1/2}$. And $\tilde{\mathbf{D}}$ is the add self loop degree matrix.

Step3:Acquiring. After L times of message passing in the tracks, we need to obtain the final representation $\mathbf{Z}$ of the paper nodes from the final messages $\mathcal{M}^L$. In this process, we requires establishing the acquiring guidance matrix $\mathbf{F}^a \in \mathbb{R}^{|\mathcal{T}|\times|\mathcal{V}|}$ to guide the computation of $\mathbf{Z}$. Specifically, $\mathbf{F}^a$ is given by

$$\mathbf{F}^a_{:,v} = \text{softmax}(\mathbf{X}\mathbf{W}_Q(\mathcal{M}^{(L)}_{:,v,:}\mathbf{W}_K)^\top), \tag{3}$$

$$\mathbf{Z}_{v,:} = (\sum\nolimits_{T\in\mathcal{T}} \mathbf{F}^a_{T,v} \cdot \mathcal{M}^{(L)}_{T,v,:})\mathbf{W}_V, \tag{4}$$

where $\mathbf{W}_:$ are learnable parameters. $\mathbf{Z}$ is used for modeling in downstream retrieval tasks.

Step4:Retrieval Task Training. This step aims to distinguish the relevant text pairs from other irrelevant or negative pairs. Given a collection of text pairs $\{(q_i, p_i)\}_{i=1}^n$, we assign a list of in-batch negative papers $\{p^-_{ij}\}_{j=1}^m$ for i-th query. Then InfoNCE loss is as follows:

$$\min L = -\frac{1}{n}\sum_i \log \frac{e^{s(q_i,p_i^+)}}{e^{s(q_i,p_i^+)} + \sum_j e^{s(q_i,p^-_{ij})}}, \tag{5}$$

where $s(q,p)$ is a scoring function between user query q and paper p. It's given by

$$s(q,p) = \cos(f_{text}(q), \mathbf{Z}_p)/\tau. \tag{6}$$

In our experiments, τ is set to 0.01. Additionally, we utilize GradCache[6] to further increase the batch size for improved performance.

3.3 Fine-tuning with Hard Negatives

During fine-tuning, we use contrastive learning to distinguish between positive and negative examples, with a key challenge being the selection of negative examples[18]. Common methods like 3.2 are often inefficient and ineffective, as in-batch negatives usually have low similarity to the query, leading to less effective training[24].

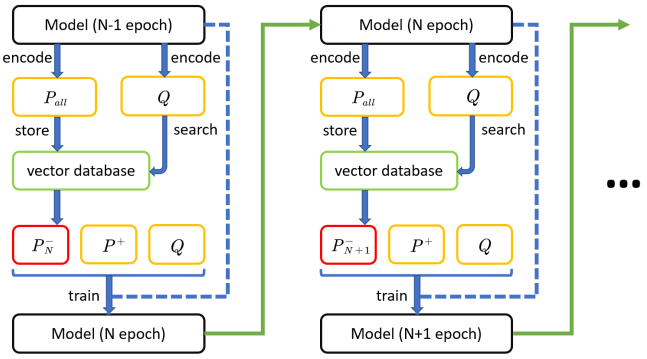

Figure 2: The framework of fine-tuning with hard negatives. P_N^- stands for hard negatives sampled in the N-th epoch.

We note that traditional hard negative mining is typically applied to an untrained model with a static dataset, ignoring the model's evolution during training. To address this, we propose a dynamic sampling approach based on hard negative mining. This approach continuously mines negative examples during training, dynamically updating the training set, ensuring the model updates gradients in the correct direction for as long as possible.

Practically, periodic hard negative mining involves mining challenging examples at intervals of several epochs (described as M). In the N-th epoch, the process is detailed in Fig.2.

Step 1: Database Building. Using the model trained up to the (N-1)-th epoch, we encode all papers P_{all} and store them in a vector database.

Step 2: Querying and Sampling. For each query Q, we encode it with the model trained up to the (N-1)-th epoch, then perform a vector similarity search against the vector database from Step 1. From the top range (e.g., top 75 to 200), we filter out the positive examples P^+, and the remaining vectors are further filtered to a specified number as hard examples P_N^-.

Step 3: Retrieval Task Training. We train the model using the newly generated dataset as defined in Section 3.2, producing the model at the N-th epoch.

Additionally, research shows that using reciprocal rank fusion (RRF)[2] to ensemble multiple models can enhance performance. Therefore, we train multiple different models and apply RRF in the final version of our method.

4 Experiments

4.1 Datasets

We use OAG-QA[25] dataset, which collected question-paper pairs from academic question-answering platforms, such as StackExchange and Zhihu. The dataset consists of a query dataset and a candidate papers list. The query dataset comprises three keywords: question, body and pids, whose values correspond to the brief description of the question, the detailed content of the question, and the list of paper identifier (pid) for the relevant papers, respectively. 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).

Query	Lable	How can I ensure convergence of DDQN, if the true Q-values for different actions in the same state are very close?
P 1st	Pos	Implicit Quantile Networks for Distributional Reinforcement Learning.
P 2nd	Neg	Full Gradient DQN Reinforcement Learning: A Provably Convergent Scheme
P 3rd	Neg	Deep Reinforcement Learning with Double Q-learning
P 4th	Neg	Quantum Computing, Postselection, and Probabilistic Polynomial-Time
P 5th	Pos	A Distributional Perspective on Reinforcement Learning.

Table 1: Case study

4.2 Case Study

Since the ground truth for the relevant test dataset has not yet been released, we will not conduct an in-depth analysis. Instead we provide some results of the queries in the valid dataset retrieved by our methods.

For the provided query, we retrieved two positive examples within the top five results, ranking first and fifth. Although the passage ranked second is not a positive example, it is highly relevant to the query. The dataset does not necessarily provide the definitive answers but rather a subset of the many possible answers. The results indicates that our method effectively learns the actual data distribution.

5 Conclusion

Our **citation-aware academic graph retriever** significantly enhances academic question answering by integrating text features and citation relationships. This system, which employs a multi-track message passing mechanism and dynamic retrieval task training, achieves high retrieval accuracy and efficiency. Our approach was validated through its competitive performance in the KDD CUP 2024 Task 2 competition, securing the 3rd place. The simplicity and effectiveness of our method make it a promising solution for academic graph mining and information retrieval tasks.

References

[1] 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. https://doi.org/10.48550/arXiv.2402.03216 arXiv:2402.03216 [cs].

[2] Gordon V. Cormack, Charles L A Clarke, and Stefan Buettcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '09)*.

[3] Julien Delile, Srayanta Mukherjee, Anton Van Pamel, and Leonid Zhukov. 2024. Graph-Based Retriever Captures the Long Tail of Biomedical Knowledge. https://doi.org/10.48550/arXiv.2402.12352 arXiv:2402.12352 [cs].

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*. 4171–4186.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. https://doi.org/10.48550/arXiv.1810.04805 arXiv:1810.04805 [cs].

[6] Luyu Gao, Yunyi Zhang, Jiawei Han, and Jamie Callan. 2021. Scaling Deep Contrastive Learning Batch Size under Memory Limited Setup. In *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*. 316–321.

[7] Xiaoxin He, Xavier Bresson, Thomas Laurent, Adam Perold, Yann LeCun, and Bryan Hooi. 2024. Harnessing Explanations: LLM-to-LM Interpreter for Enhanced Text-Attributed Graph Representation Learning. https://doi.org/10.48550/arXiv.2305.19523 arXiv:2305.19523 [cs].

[8] Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh V. Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. 2024. G-Retriever: Retrieval-Augmented Generation for Textual Graph Understanding and Question Answering. https://doi.org/10.48550/arXiv.2402.07630 arXiv:2402.07630 [cs].

[9] Jin Huang, Xingjian Zhang, Qiaozhu Mei, and Jiaqi Ma. 2024. Can LLMs Effectively Leverage Graph Structural Information through Prompts, and Why? https://doi.org/10.48550/arXiv.2309.16595 arXiv:2309.16595 [cs].

[10] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. https://doi.org/10.48550/arXiv.2005.11401 arXiv:2005.11401 [cs].

[11] Jiachang Liu, Dinghan Shen, Yizhe Zhang, William B Dolan, Lawrence Carin, and Weizhu Chen. 2022. What Makes Good In-Context Examples for GPT-3?. In *Proceedings of Deep Learning Inside Out (DeeLIO 2022)*. 100–114.

[12] Linhao Luo, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. 2024. Reasoning on Graphs: Faithful and Interpretable Large Language Model Reasoning. https://doi.org/10.48550/arXiv.2310.01061 arXiv:2310.01061 [cs].

[13] Ana Marasović, Iz Beltagy, Doug Downey, and Matthew E Peters. 2022. Few-Shot Self-Rationalization with Natural Language Prompts. In *Findings of the Association for Computational Linguistics: NAACL 2022*. 410–424.

[14] Sewon Min, Danqi Chen, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2020. Knowledge Guided Text Retrieval and Reading for Open Domain Question Answering. https://doi.org/10.48550/arXiv.1911.03868 arXiv:1911.03868 [cs].

[15] Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Ábrego, Ji Ma, Vincent Y. Zhao, Yi Luan, Keith B. Hall, Ming-Wei Chang, and Yinfei Yang. 2021. Large Dual Encoders Are Generalizable Retrievers. https://doi.org/10.48550/arXiv.2112.07899 arXiv:2112.07899 [cs].

[16] Hongbin Pei, Yu Li, Huiqi Deng, Jingxin Hai, Pinghui Wang, Jie Ma, Jing Tao, Yuheng Xiong, and Xiaohong Guan. 2024. Multi-Track Message Passing: Tackling Oversmoothing and Oversquashing in Graph Learning via Preventing Heterophily Mixing. In *Forty-first International Conference on Machine Learning*. https://openreview.net/forum?id=1sRuv4cnuZ

[17] Chen Qian, Huayi Tang, Zhirui Yang, Hong Liang, and Yong Liu. 2023. Can Large Language Models Empower Molecular Property Prediction? https://doi.org/10.48550/arXiv.2307.07443 arXiv:2307.07443 [cs, q-bio].

[18] Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. 2020. Contrastive learning with hard negative samples. *arXiv preprint arXiv:2010.04592* (2020).

[19] Weng Tam, Xiao Liu, Kaixuan Ji, Lilong Xue, Jiahua Liu, Tao Li, Yuxiao Dong, and Jie Tang. 2023. Parameter-Efficient Prompt Tuning Makes Generalized and Calibrated Neural Text Retrievers. In *Findings of the Association for Computational Linguistics: EMNLP 2023*. 13117–13130.

[20] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. 2008. Arnetminer: extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. 990–998.

[21] Heng Wang, Shangbin Feng, Tianxing He, Zhaoxuan Tan, Xiaochuang Han, and Yulia Tsvetkov. 2024. Can Language Models Solve Graph Problems in Natural Language? https://doi.org/10.48550/arXiv.2305.10037 arXiv:2305.10037 [cs].

[22] 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. https://doi.org/10.48550/arXiv.2212.03533 arXiv:2212.03533 [cs].

[23] Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Improving Text Embeddings with Large Language Models. https://doi.org/10.48550/arXiv.2401.00368 arXiv:2401.00368 [cs].

[24] Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021. Optimizing dense retrieval model training with hard negatives. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1503–1512.

[25] 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*.

[26] Haiteng Zhao, Shengchao Liu, Chang Ma, Hannan Xu, Jie Fu, Zhi-Hong Deng, Lingpeng Kong, and Qi Liu. 2023. GIMLET: A Unified Graph-Text Model for Instruction-Based Molecule Zero-Shot Learning. http://arxiv.org/abs/2306.13089 arXiv:2306.13089 [cs, q-bio].