# Hybrid Retrieval Systems Based on LLMs Embedding and Enhancement

Qinglun Wang
Onewo Space-Tech Service Co., Ltd.
Shenzhen, Guangdong, China
wangql43@vanke.com

Ji Yuan*
Onewo Space-Tech Service Co., Ltd.
Shenzhen, Guangdong, China
yuanj36@vanke.com

Xinlong Huang
Shenzhen University
Shenzhen, Guangdong, China
2210295079@email.szu.edu.cn

Jia Yan
Onewo Space-Tech Service Co., Ltd.
Shenzhen, Guangdong, China
yanj93@vanke.com

Rixin Xiao
Onewo Space-Tech Service Co., Ltd.
Shenzhen, Guangdong, China
xiaorx01@vanke.com

Xianfeng Ding
Onewo Space-Tech Service Co., Ltd.
Shenzhen, Guangdong, China
dingxf07@vanke.com

## Abstract

With the increasing popularity of large language models (LLMs), the retrieval of accurate and effective scientific research documents has become crucial in enhancing the ability of language models to answer user questions. To address this challenge, the AQA-KDD-2024 competition is launched by Tsinghua University's Knowledge Engineering Group (KEG), in collaboration with ZhipuAI. In the competition, the Onewo algorithm team has developed an innovative approach consisting of four stages: i.e. data processing and enhancement with LLMs, candidate generation, ranking candidates, and weighted ensemble. A key highlight of our approach is data enhancement, where the LLMs model is utilized to improve query and body texts. This involves generating keywords and providing AI responses based on an effective prompt template. Through our test benchmark, we have achieved a significant improvement in the performance metric. The score has progressed from 0.16526 to an enhanced score of 0.18367. With our innovative solution, our team Onewo won the 8th place in the final leaderboard of the AQA-KDD-2024 competition. The code is available at this link: https://github.com/Starrylun/AQA-KDD-2024-Rank8.

## CCS Concepts

• **Computing methodologies** → Large language Model.

## Keywords

Retrieval System, Data Enhancement, Candidate Generation, Ranking Candidates

**ACM Reference Format:**
Qinglun Wang, Ji Yuan, Xinlong Huang, Jia Yan, Rixin Xiao, and Xianfeng Ding. 2024. Hybrid Retrieval Systems Based on LLMs Embedding and Enhancement. In *Proceedings of ACM KDD AQA-OAG-Challenge (Conference KDD '2024).* ACM, Barcelona, Spain, 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
*Conference KDD '2024, 2024, Barcelona, Spain*
© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/10.1145/nnnnnnn.nnnnnnn

## 1 Introduction

Increasingly large language models (LLMs) are being employed to support various domain-specific tasks. Indeed, fields such as law, finance, and healthcare have consistently been among those receiving high levels of attention. Nevertheless, the attention given to the field of scientific research is not high, and thus providing high-quality, cutting-edge academic knowledge across multiple fields for researchers and the general public has become an urgent matter. Therefore, this competition is held with the aim of letting developers design a retrieval system that retrieve the most relevant papers from a given pool of candidates in response to professional questions [1].

Our team, utilizing experience in domain-specific search, achieved eighth place in the competition. In Figure 1, our solution approach comprises several key strategies:

(1) Data processing and enhancement with LLMs;
(2) Candidate generation with retrieval models;
(3) Re-ranking models with LLMs' Embedding;
(4) Weighted Ensemble.

## 2 Retrieval Models

### 2.1 MiniLM

The MiniLM model represents an efficient knowledge distillation technique for reducing the size of extensive pre-trained Transformer-based language models. During training, the student model deeply mimics the self-attention mechanisms of the teacher model, which are integral to the Transformer architecture. Researchers introduce a novel approach that leverages the self-attention distributions and value relationships from the teacher model's final Transformer layer to direct the student's training. This method has proven to be both effective and adaptable for various student models [2].

### 2.2 BAAI General Embedding

BAAI general embedding (BGE) is a general Embedding Model. Authors pre-train the models using retromae, and a series of bge models are trained on large-scale pair data by using contrastive learning method. The primary objective of BGE is to mitigate the constraints encountered by large models in real-world applications, particularly within the realms of natural language content retrieval, comprehension, and generation. The design philosophy of this model is centered around delivering a universal, high-impact

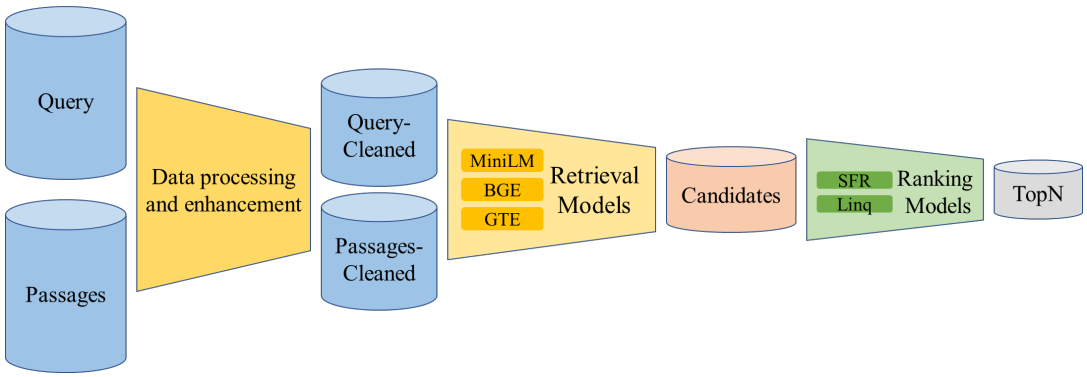


**Figure 1: The overall pipeline**

Chinese-English domain semantic vector solution, which not only excels in semantic retrieval precision but also surpasses contemporary community benchmarks in terms of overall semantic representation capabilities [3].

The focus of this competition's dataset is primarily on scientific research, with the vast majority of the information being in the context of English language corpora. Therefore, according to the massive text embedding benchmark (MTEB), we have chosen bge-v1.5-en-large and bge-m3.

### 2.3 General Text Embeddings

General Text Embeddings (GTE) has proven to be an effective strategy for textual representation. The Dual Encoder framework is typically adopted by the text representation model trained based on supervised data. In this architecture, both query and passages are first encoded by a pre-trained language model (PLM). Subsequently, the vector extracted from the [CLS] token position within the PLM is typically employed as the ultimate representation of the text. Therefore, we use a model, named nlp-gte-sentence-embedding-english-large, to enhance different representation effects [4].

## 3 Ranking Models

### 3.1 SFR Embedding Mistral

The SFR-Embedding-Mistral is constructed on the solid foundations of E5-mistral-7b-instruct and Mistral-7B-v0.1. Benefiting from the powerful text comprehension and representation capabilities of the Mistral model, it demonstrates enhanced generalization abilities through training on diverse datasets from different tasks such as clustering, classification, and semantic text similarity [5]. Compared to previous models, it has shown substantial improvements in retrieval performance, ranking among the top on the MTEB.

The effectiveness of the model can be attributed to several aspects:

(1) During the model training stage, the researchers employed LoRA adapter to fine-tune the E5-mistral-7b-instruct model;
(2) The adaptability and performance of models are enhanced by transfer learning from multiple tasks;
(3) Task-homogeneous batching increases the difficulty of the contrastive objective for the model, promoting enhanced generalization;

### 3.2 Linq Embed Mistral

Linq-Embed-Mistral adopts the fundamental architecture from SFR-Embedding-Mistral. The researchers' primary objective is to enhance the efficiency of text retrieval through the employment of sophisticated data refinement techniques, such as data cleaning, filtering and hard negative mining by teacher models, to improve the quality of the LLM synthetic data.

## 4 Related Works

### 4.1 Dataset Description

The initial step in modeling is to acquire a comprehensive understanding of our dataset. The OAG-QA dataset predominantly encompasses domains within the natural and engineering sciences. The dataset is mainly divided into four parts in Table 1 :

**Table 1: Dataset description**

| Type | File name | Fields |
|---|---|---|
| Train dataset | qa-train.txt | question, body, pids |
| valid dataset | qa-valid-wo-ans.txt | question, body |
| test dataset | qa-valid-wo-ans.txt | question, body |
| documents | pid-to-title-abs-new.json | title, abstract, pid |

Below are specific explanations:

- `question`: the specific query or research problem posed by the user.
- `body`: the content and explanation of the question.
- `pid`: the unique id of the paper from documents.
- `pids`: the list of each pid.
- `title`: the name of the paper from documents.
- `abstract`: the most important facts or ideas of paper.

### 4.2 Data Processing

An essential factor contributing to the recent achievements of large-scale language models lies in the employment of vast and continually growing textual datasets for unsupervised pre-training purposes. Consequently, a critical initial step in both the training and inference phases of language models is filtering out trash data, which can improve the quality of training models [6].

Based on the assertions in the [7] , substantial amounts of irregular text information, such as HTML tags, consecutive spaces, and line breaks, can negatively impact a model's training and inference processes. Given that the raw body field contains a plethora of HTML-formatted text, we have devised a straightforward set of rules to process text extracted from web scraping or containing HTML tags and special characters.

(1) This step employs regular expressions to eliminate HTML tags from the text;
(2) It removes sequences of multiple spaces and line breaks;
(3) Leading and trailing white-space characters, including spaces and tabs, are stripped from the strings to make the text more compact;
(4) Specific URL-related strings, such as "http://", "https://", ".com", and ".cn", are also removed.

## 4.3 Data Enhancement with LLMs

Data augmentation for text is an effective strategy to tackle the challenges posed by the scarcity and inferior quality of samples in many Natural Language Processing (NLP) tasks [8]. Especially, in the field of text search, query expansion is a widely used technique that improves the recall of search systems by adding additional terms to the original query. The expanded query may be able to recover relevant documents that have no lexical overlap with the original query [9].

Inspired by the recent successes of LLMs, particularly the development of ChatGPT, which has demonstrated enhanced language understanding capabilities, there is a growing interest in advancing this field. In [10, 11], researchers expand the original query with LLM output in order to help during document retrieval. Because, LLMs are not restricted to the initial retrieved set of documents and may be able to generate expansion terms not covered by traditional methods.

In our work, we deploy two models to perform query enhancement tasks. These models are:

(1) Qwen1.5-14b-gptq-int4;
(2) GLM4-9b-chat.

Two enhancement approaches are adopted:

(1) Generate the keywords of query;
(2) Model response of the query.

The strategy for prompts involves utilizing the context-objective-style-tone-audience-response (CO-STAR) template, guiding the models to output enriched text content according to specified instructions. Specifically, incorporating CO-STAR formatted inputs enables AI to better comprehend the focal points of your queries. Analytically, all large language model-based AI systems can leverage this approach, thereby enhancing their capability to process information more efficiently. This results in more precise responses to your queries and facilitation of superior ideas, as it ensures the AI is aligned with the core intent behind your questions.

Meanwhile, we have deployed two sets of models using vLLM to facilitate rapid inference. The Figure 2 shows the entire data augmentation process.

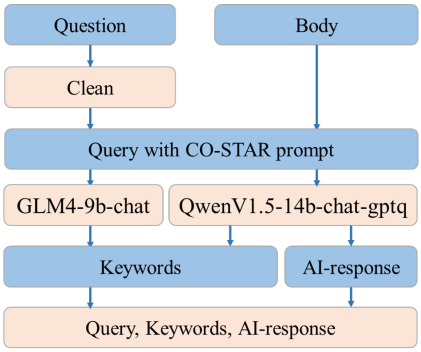


**Figure 2: The pipeline of data enhancement with LLMs**

In Table 2 , the experimental results of test benchmark indicate that when recall and re-ranking models remain constant, there is a progressively increasing trend in the scores as query augmentation based on LLMs is incrementally introduced.

## 4.4 Retrieval and Ranking Methods

Within NLP research domains, model ensemble is a prevalent technique that combines predictions from multiple models to enhance overall predictive performance. This approach proves effective because different models can learn distinct patterns from the data, and when their predictions are collectively considered, they can complement one another, reducing errors and thereby improving algorithm accuracy and stability [1].

In our works, we have chosen two models of MiniLM, namely all-MiniLM-L6-v2 and all-MiniLM-L12-v2. After the data processing is completed, we concatenate the `question` and `body` using the `\n` symbol as the delimiter to form the `query`, and join the `title` and `abstract` together to create the `passages`. Meanwhile, we encode the query and passage using 2 models, normalize the encoded vectors, calculate the dot product of the two vectors, and then select the top 150 candidates.

Furthermore, in the stage of encoding query by using BGE and GTE models, we clean the original text data, fill the 'query' and 'body' fields into the 'Represent this sentence for searching relevant passages: {query} {body}', and regard the sentence as user query. In the process of handling the passage, the title and abstract information are also incorporated into 'Represent this sentence for searching relevant passages: {title} {abstract}' to ensure consistency. The query and passage are encoded by BGE models. Moreover, in our study, it is important to highlight that the BGE and GTE models utilize the last hidden state of [CLS] to represent the sentence embedding.

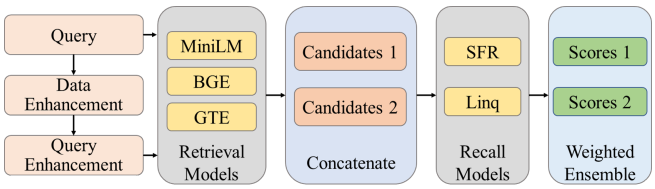


**Figure 3: Weighted ensemble method**

**Table 2: LLMs-based query expansion in test leaderboard**

| Model | Method | MAP@20 |
|---|---|---|
| Base structure of test | No Keywords, No AI-response | 0.16526 |
| + Qwen1.5-14b-gptq-int4 | Keywords | 0.17554 |
| + Qwen1.5-14b-gptq-int4 | Keywords, AI-response | 0.18342 |
| **+ GLM4-9b-chat** | Keywords | **0.18367** |

Based on the scores of online validation benchmark, we employ the weighted ensemble method to combine the outputs from both the recall and ranking stages. As depicted in the illustration in Figure 3, many candidates are retrieved by some recall models. These candidates are then consolidated and scored by ranking models, including SFR-Embedding-Mistral and Linq-Embed-Mistral. Subsequently, ensemble operations at the coefficient level are performed, taking into account various inputs and output results.

## 4.5 Experiments

For each question $V_q$ , the Average Precision (AP) will be calculated according to the following formula:

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

Here, $R_q$ is the number of positive paper IDs, $M$ denotes the total number of corpus papers, and $P_q(k)$ represents the precision up to the $k$-th item in the ranked list for question $V_q$. $\mathbf{1}_k$ is an indicator function; if the $k$-th returned paper preference meets the standard answer, then $\mathbf{1}_k = 1$, otherwise $\mathbf{1}_k = 0$.

For a given set of $n$ questions, we calculate the Mean Average Precision (MAP) as follows:

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

While choosing models, we conduct some experiments on the validation leaderboard. According to the online evaluation metrics, we decide which recall models to use.

The table 3 primarily delineates improved scores of validation after being processed by some models. Therefore, we regard weighted ensemble models as the base structure of test, which is applied to the inference the test task. The changes in MAP@20 are shown in Table 2 .

**Table 3: Top20 score in validation leaderboard**

| Model | MAP@20 |
|---|---|
| random selection | 4.4169e-5 |
| all-MiniLM-L6-v2 | 0.14468 |
| + bge-v1.5-en-large | 0.15926 |
| + all-MiniLM-L12-v2 | 0.15941 |
| + SFR-Embedding-Mistral | 0.19486 |
| + Linq-Embed-Mistral | 0.20205 |
| + Weighted Ensemble | **0.20237** |

## 5 Conclusion

The AQA-KDD-2024 competition presents a distinctive opportunity as it specifically targets the domain of academic paper retrieval. Our solution is a hybrid ensemble system for retrieval and ranking, leveraging open-source models. Here are some key highlights:

(1) Query expansion with Qwen and GLM4;
(2) Rapidly generating candidates by retrieval models;
(3) Accurately sorting candidates by ranking models with LLMs embedding;
(4) Improving performance of hybrid models by weighted ensemble.

While our work boasts several strengths, it is important to acknowledge its limitations. Notably, we have yet to fully leverage retrieval models of LLMs during recall phase, relying instead on traditional methods. Finally, our team has won the 8th place in the final leaderboard, demonstrating the effectiveness of our approach while highlighting areas for the future improvement.

## References


[1] Chris Deotte, Jean-Francois Puget, Benedikt Schifferer, Gilberto Titericz, et al. Winning amazon kdd cup'23. In *Amazon KDD Cup 2023 Workshop*, 2023.

[2] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems*, 33:5776–5788, 2020.

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

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

[5] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

[6] Alon Albalak, Yanai Elazar, Sang Michael Xie, Shayne Longpre, Nathan Lambert, Xinyi Wang, Niklas Muennighoff, Bairu Hou, Liangming Pan, Haewon Jeong, et al. A survey on data selection for language models. *arXiv preprint arXiv:2402.16827*, 2024.

[7] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[8] Tianjun Zhang, Shishir G Patil, Naman Jain, Sheng Shen, Matei Zaharia, Ion Stoica, and Joseph E Gonzalez. Raft: Adapting language model to domain specific rag. *arXiv preprint arXiv:2403.10131*, 2024.

[9] Mingrui Wu and Sheng Cao. Llm-augmented retrieval: Enhancing retrieval models through language models and doc-level embedding. *arXiv preprint arXiv:2404.05825*, 2024.

[10] Rolf Jagerman, Honglei Zhuang, Zhen Qin, Xuanhui Wang, and Michael Bendersky. Query expansion by prompting large language models. *arXiv preprint arXiv:2305.03653*, 2023.

[11] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35, 2023.