Enhancing Biomedical Question Answering with Parameter-Efficient Fine-Tuning and Hierarchical Retrieval Augmented Generation

Yichen Gao[1,†], Licheng Zong[1,†] and Yu Li[1,2,*]

[1] *The Chinese University of Hong Kong, Hong Kong SAR, 999077, China*
[2] *The CUHK Shenzhen Research Institute, Hi-Tech Park, Shenzhen, 518057, China*

Abstract

This paper reports the work done by the CUHK-AIH team in the 12th BioASQ Challenge task 12b, which involves Phases A, A+, and B. In Phase A, we build BM25 indexes for all documents from PubMed Central (PMC). When an input question is received, the system uses the question as a keyword to retrieve relevant documents from PMC via the BM25 retriever, obtaining a list of targeted documents. For Phase A+, we construct a hierarchical Retrieval-Augmented Generation (RAG) pipeline based on the Llama2-chat-7B model. The model is fine-tuned on the BioASQ training set using a Parameter-efficient Fine-Tuning (PEFT) method called Low-Rank Adaptation (LoRA). The system further refines the search results from Phase A by employing an ensemble retriever that combines sparse and dense retrievers to identify the most relevant chunks. Finally, the system feeds the question and the most relevant chunks into the base model to generate the answer using appropriate prompts. In Phase B, the answer generation pipeline is similar to Phase A+, with the main difference being that we directly build indexes for the questions and their relevant snippets, treating snippets as the basic retrieval unit. We conducted detailed ablation studies and analyses on the model types and retrieval techniques, which indicate that PEFT and RAG can significantly improve the performance in biomedical Question Answering (QA) tasks.

Keywords

Biomedical Question-Answering, Retrieval-Augmented Generation, Large Language Model, Parameter-Efficient Fine-Tuning, BioASQ

1. Introduction

In recent years, Large Language Models (LLM) have undergone significant development and have gained widespread adoption [1, 2, 3, 4, 5], particularly in the biomedical domain including medical informatics [6], medical imaging [7], and bioinformatics [8, 9, 10]. To address the requirements of healthcare professionals and medical education, researchers have embarked on investigating the utilization of Large Language Models (LLM) for medical knowledge queries. Their objective is to integrate intricate medical knowledge into LLMs, thereby providing a knowledge query system with more accurate and comprehensive medical information. Techniques like In-context Learning [11] and Retrieval Augmented Generation (RAG) [12] have been proposed, which introduces a novel paradigm for tackling knowledge queries and responses within specialized domains.

In the field of medical information retrieval, a critical issue lies in constructing a robust information retrieval system to effectively manage a massive corpus of medical literature [13]. BioASQ [14] presents a platform to address this challenge, encouraging researchers to develop better intelligent retrieval systems. This paper focuses on task 12b [15], particularly the information retrieval, and the question-answering tasks, and tackles Phase A, Phase A+, and Phase B. Phase A involves finding relevant documents or snippets to a biomedical question. Phase B provides biomedical inquiries alongside relevant snippets (one or several sentences) and tasks participants with generating either the exact or

CLEF 2024: Conference and Labs of the Evaluation Forum, September 09–12, 2024, Grenoble, France

[*] Corresponding author.

[†] These authors contributed equally.

ycgao23@cse.cuhk.edu.hk (Y. Gao); lczong21@cse.cuhk.edu.hk (L. Zong); liyu@cse.cuhk.edu.hk (Y. Li)
https://lczong.com/ (L. Zong); https://liyu95.com/ (Y. Li)
0000-0001-5418-005X (L. Zong); 0000-0002-3664-6722 (Y. Li)

ideal answers based on these snippets. Phase A+, which is a new phase this year, will directly measure the performance of the system while providing only the biomedical question. It's worth noting that in this work, we only focus on the ideal answer part and will not be involved in the evaluation of the exact answer.

Here we utilize the BM25 retriever to search documents in Phase A, and the Parameter-Efficient Fine-Tuning on the BioASQ corpus [16] to generate high-quality answers in Phase B. In Phase A+, we propose a hierarchical Retrieval-Augmented Generation pipeline combined with the mentioned BM25 retriever and PEFT for searching and generation. So, our system is called **C**orpus **P**EFT **S**earching (CPS). For convenience, we defined the systems we used in Phase A, B, and A+ as CPS-A, CPS-B, and CPS-A+ respectively. By illustrating the results of our system and conducting detailed ablation studies, we conclude that the PEFT and hierarchical RAG bring significant improvement on the model performance.

2. Related Work

Since this paper can be divided into pipeline-related research and retrieval-related research, this section will outline two parts of related work in Section 2.1 and Section 2.2. Specifically, we will begin by sorting out the related work of the main pipeline, which consists of LLMs in medical informatics, parameter-efficient model fine-tuning, and retrieval augmented generation. After that, we will introduce the structure of the retrieved text in RAG, followed by a work that converts the retrieved text composition.

2.1. Construction of Pipeline: LLM, PEFT, and RAG

2.1.1. Large Language Models (LLMs) in Medical Informatics

In recent years, the advancement of Large Language Models (LLMs), exemplified by GPT[1], BERT[2] and Llama[3, 4], alongside the abundant public medical text data from platforms like PubMed[1], has encouraged researchers to explore diverse applications of LLMs in medical informatics. Agrawal et al. [11] reveals that sizable language models, even without explicitly training for medical fields, exhibit the capacity to extract clinical information in few-shot settings. The latest iteration, Med-PaLM2 [6], fine-tuned from Flan-PaLM [17], attained an accuracy of 0.865 on the MedQA [18] benchmark designed for generating biomedical LLMs. In addition, Venigalla et al. [19] further conducted full-parameter training of the model using data from PubMed, enhancing the model's capabilities in medical question-answering. Despite the apparent complexity of training models with a substantial number of parameters, the application of Parameter-Efficient Fine-Tuning (PEFT) techniques such as Low-Rank Adaptation (LoRA) [11] has empowered researchers to efficiently fine-tune various models both in English and Chinese medical context [20, 21, 22, 23], including our work.

2.1.2. Parameter-Efficient Fine-Tuning (PEFT)

Parameter-Efficient Fine-Tuning (PEFT) techniques encompass a series of fast, effective, and low GPU computing and memory demand model fine-tuning methods. They are widely applied in the research of Large Language Models, especially in specific domain applications [22, 23]. Adapter tuning [24] involves introducing small neural network modules (adapters) into the Transformer model and using a bottleneck architecture to compress and restore the original feature vectors, adapting the language model to new tasks. Prefix Tuning [25], proposed by X. L. Li and P. Liang, arranges a series of trainable prefixes for each Transformer layer to improve performance in specific domains. Meanwhile, by freezing the original matrix and approximating the parameter updates through low-rank decomposition, Low-Rank Adaptation (LoRA) [11] fine-tunes models with efficient memory and storage usage.

[1] https://pubmed.ncbi.nlm.nih.gov

2.1.3. Retrieval Augmented Generation (RAG)

To supplement the knowledge required by large language models when handling knowledge-intensive tasks, the system can retrieve information from large corpora like Wikipedia to generate prompt information [12]. The retrieval process can be performed using sparse retrieval, utilizing methods such as BM25 [26] or through dense retrieval [27] using vector similarity based on embedding models like Bge [28]. To capture text relationships and knowledge across multiple documents, recent work [29, 30, 31] treats multiple documents as a graph and conducts subgraph retrieval within the graph. There are also applications and frameworks, such as ChatPDF [2] and LangChain [3], parsing large segments of prompt information for question-answering systems by inputting and summarizing entire documents in segmented fashion, thereby augmenting knowledge.

In recent years, RAG also gains much attention in the biomedical field. MedCPT [32] is proposed to generate better biomedical sentence representations for zero-shot semantic information retrieval. It consists of a pair of Transformer-based retriever and re-ranker pre-trained on PubMed user click logs by contrastive learning. Xiong et al. [33] proposed MIRAGE and MEDRAG to conduct systematic and large-scale experiments in the field of medical RAG, which shows that RAG based on biomedical corpus can improve the performance of GPT-3.5 and Mixtral to the level of GPT-4.

2.2. Retrieval Units: Chunk and Snippet

2.2.1. Chunk: Generally Used Basic Unit of Retrieval

Recently, numerous studies have focused on tasks within the medical domain, enhancing the capacity of Large Language Models (LLMs) to tackle Question Answering (QA) tasks related to medical knowledge through the application of Retrieval Augmented Generation (RAG) on biomedical databases. In these studies, due to the substantial size of their databases and the absence of structured organization or sentence extraction methods for Question Answering inquiries, the entirety of text or chunks is employed as the basic unit of retrieval. GeneGPT[34] obtains knowledge for LLM in the genetic field by calling the web API provided by the National Center for Biotechnology Information (NCBI). When GeneGPT organizes this knowledge, it uses the Codex [35] model with long context length (8k tokens) to handle structured API return results. Based on the Chain of Thought [36], the problem that requires multiple returns from the API is decomposed into multiple sub-problems. The RAG pipeline that uses a vectorized database, such as Chat-Orthopedist [37], splits the complete document in the database into chunks with the same length, which has a suitable length for retrieval and input to LLM. In addition, for RAG tasks in non-medical fields, approaches dividing documents into chunks are also very typical. Usually, vector databases such as chromaDB [4] and Pinecone [5] are employed as the basis for construction.

2.2.2. Snippet: Basic Unit of Retrieval in BioASQ

Besides the exemplary medical knowledge Question Answering Large Language Model (LLM) pipelines, which stand out in the research domain, we also examined other systems featured in the BioASQ task 11b's publication (CLEF-WN 2023 [38]). These systems primarily utilize the provided snippets directly, but there are some slight variations in specific applications. UR-gpt [39], which performs very well in the rankings, inserts snippets directly into prompts for in-context learning. The Sentence-based Ranking [40] system's method is close to the previous system, but they used the top 5 most relevant sentences due to the input length limit of its model. The IISR [41] firstly selects the top 5 most relevant snippets and uses ChatGPT to truncate or summarize them according to a fixed length. The context is input as the ChatGPT conversation history instead of fusing it together with the question in IISR.

[2] https://www.chatpdf.com

[3] https://www.langchain.com/

[4] https://www.trychroma.com/

[5] https://www.pinecone.io/

3. Methodology

This section details the pipelines we employed in each phase. Our system is called **C**orpus **P**EFT **S**earching (CPS), for our main contribution is to integrate Parameter-Efficient Fine-Tuning (PEFT) on the PMC corpus with a hierarchical retrieval-based searching method in the BioASQ 12b task. Since the inputs and targeted outputs for submission are different in the three phases, we built three pipelines named CPS-A, CPS-B, and CPS-A+ for Phase A, B, and A+ submissions respectively. Table 1 shows the systems used in different phases and the sections and figures that describe them.

In Phase A (Section 3.1) we leveraged the BM25 retriever to search relevant documents and return the document list. Then in Phase B (Section 3.2), we used the BioASQ training set as the corpus to fine-tune a Large Language Model (LLM) based on a Parameter-Efficient Fine-Tuning (PEFT) method to generate more reasonable answers. Finally, in Phase A+ (Section 3.3), we proposed a hierarchical Retrieval-Augmented Generation (RAG) method combined with the methods mentioned above to construct a general biomedical Q&A system, which could serve as a chatbot for any other biomedical questions.

Table 1
The systems used in different phases with corresponding sections and figures that describe them.

System	Section	Figure	Phase
CPS-A	Section 3.1	Figure 1	Phase A
CPS-B	Section 3.2	Figure 2	Phase B
CPS-A+	Section 3.3	Figure 3	Phase A+

3.1. Phase A

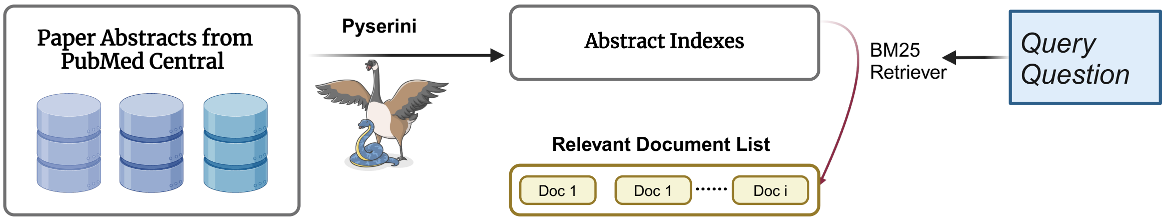

Figure 1: Overview of system CPS-A (**C**orpus **P**EFT **S**earching - phase **A**), the retrieval pipeline for BioASQ task 12b phase A.

As illustrated in Figure 1, in Phase A, we first downloaded all the abstracts available from PubMed Central (PMC)[6] and employed the package Pyserini[42] to build BM25 indexes of these abstracts, treating each abstract as an individual document. Subsequently, we applied the BM25 retriever to search related documents using the query questions as the keywords. After being filtered by an appropriate similarity score threshold, the final list of relevant articles is obtained.

3.2. Phase B

Our work in Phase B can be divided into two parts: **Pre-processing** and **Retrieval & Generation**, as shown in Figure 2. In the Pre-processing stage, we built BM25 indexes of the golden enriched relevant snippets using Pyserini[42] for later further retrieval. To help generate more accurate answers, we used a Parameter-Efficient Fine-Tuning (PEFT) method, LoRA[43] to fine-tune the Llama2-chat-7B model[4] on the BioASQ corpus, which is the Question-Answering pairs from BioASQ task 12b training

[6] https://www.ncbi.nlm.nih.gov/pmc/

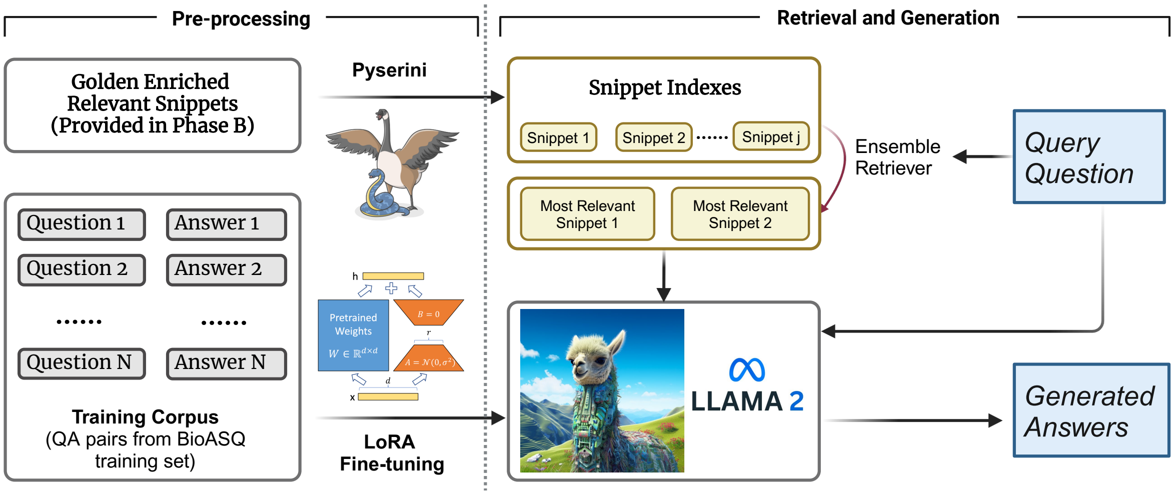

Figure 2: Overview of system CPS-B (**C**orpus **P**EFT **S**earching - phase **B**), the proposed LLM pipeline with parameter-efficient fine-tuning (PEFT) for BioASQ task 12b phase B.

set (consisting of 5049 Q&A pairs). Due to the computing resource limit, we were unable to directly fine-tune the whole Large Language Model on the corpus. The Low-Rank Adaptation (LoRA)[43] helps reduce the fine-tuning time and resources significantly while ensuring the model keeps a satisfying performance. The work in this stage was done before the test sets were released and wouldn't be repeated later.

The Retrieval & Generation stage is the process to generate answers for the query questions in Phase B. Our system leverages an ensemble retriever combining sparse (BM25) and dense (vector similarity) retrievers for searching the most relevant snippets and provides them as references to the LLM. Bge[7] was used to embed the chunks into vectors for similarity searching. The reason we constructed the retriever is that we found the performance of directly inputting all snippets into the model are not good enough, which we will discuss in Section 4.3.3.

Given appropriate prompts with the retrieved snippets and the query question, the fine-tuned model can generate a response for the query question to be submitted in Phase B. The prompt template is shown below, which we modified from Alpaca[8] by adding a role definition at the beginning.

```
You are an expert in the field of biomedical science.
Below is an instruction that describes a task, paired with an input that provides further context. Write a response that appropriately completes the request.
### Instruction: {question}
### Input: {context}
### Response:
```

In this template, the position of 'context' will be replaced by the relevant snippets found in the retrieval procedure, while 'question' will be replaced by the query question body.

3.3. Phase A+

Phase A+ is more challenging than Phase B since there are no golden-enriched snippets. Therefore, our system needs to first search the relevant documents and snippets from the whole PMC database, and then generate answers. Here, we implement a hierarchical Retrieval Augmented Generation (RAG) pipeline combining the insights from Phase A and B (shown in Figure 3).

[7] https://huggingface.co/BAAI/bge-large-en
[8] https://github.com/tatsu-lab/stanford_alpaca

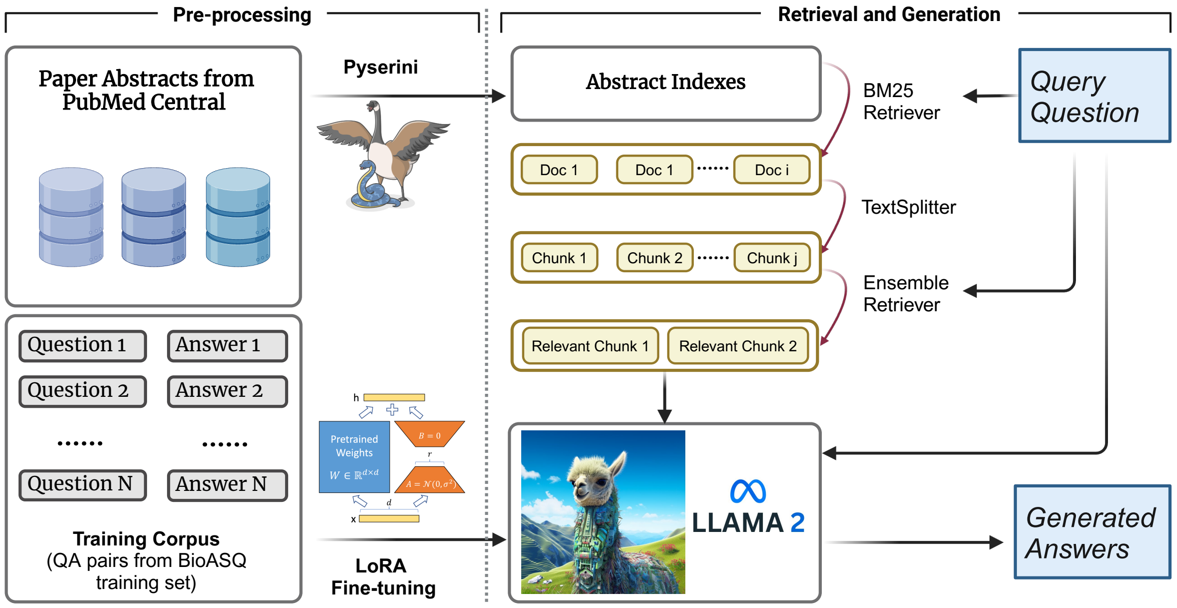

Figure 3: Overview of system CPS-A+ (**C**orpus **P**EFT **S**earching - phase **A+**), the proposed LLM pipeline with fine-tuning and hierarchical retrieval augmented generation for BioASQ task 12b phase A+.

Similarly to Phase B, the framework consists of two stages: **Pre-processing** and **Retrieval & Generation**. The fine-tuned model in the Pre-processing stage is identical to that in Phase B, while the indexes are built from the whole PMC abstracts rather than golden-enriched snippets.

In the Retrieval & Generation stage, we propose a hierarchical strategy to search question-relevant texts. Specifically, considering the quicker nature of sparse retrieval compared to dense retrieval, we still employed the BM25 sparse retrieval indexed on PMC abstracts as the first-level retrieval. This process selects the most relevant abstracts as the preliminary information screening. These documents are then segmented into multiple chunks by the TextSplitter from LangChain Package[9] for the next search. Subsequently, an ensemble retriever which is the same as the retriever in phase B is employed as a more precise second-level retrieval. This ensemble retriever will identify multiple chunks most relevant to the question as the second level of retrieval. Finally, these most relevant chunks and the questions are combined by the prompt template mentioned in Section 3.2 to form the model input.

4. Results and Analyses

4.1. Experiment Settings

4.1.1. Datasets

In the official submissions this year, we utilize the training set of BioASQ Task 12b (consisting of 5049 question-answer pairs) as the LoRA fine-tuning (mentioned in Section 3.2 and Section 3.3) dataset for CPS-B and CPS-A+ submissions. In the ablation study experiments in Section 4.3, we employ the training set of BioASQ Task 11b (consisting of 4719 question-answer pairs) as the LoRA fine-tuning dataset and the test set of BioASQ Task 11b batch 1 (consisting of 90 question-answer pairs) as the test set to do the evaluations.

4.1.2. Model Deployment

Considering factors such as deployment complexity, open-source availability, and parameter scale, we chose the Llama2-7b-chat mode as the basic model in this work. For not focusing on the model

[9] LangChain:langchain_text_splitters.RecursiveCharacterTextSplitter

itself, this study adheres to default settings for model parameters and fine-tuning parameters. In all our experiments and submissions, we use fp16 precision for both model fine-tuning and inference. We performed our experiments on a Linux server with two NVIDIA GeForce RTX 3090 GPUs.

4.2. BioASQ Task 12b Official Evaluations

In this section, we present the official evaluation results of our system in the four batches of BioASQ Task 12b. For comparison, we have included the best-performing systems in each batch. As there are multiple metrics, we selected the system that ranked in the top 5 among all metrics in each batch. If no system met this criterion, we expanded the selection to include systems ranked in the top 10. The selected system is referred to as the "Top Competitor" in the tables in Section 4.2.

4.2.1. Phase A

We utilized the BM25 retrieval-based CPS-A system to retrieve relevant documents in phase A (introduced in Section 3.1). Here we show the submission results of the retrieved document in Table 2. As for selecting the relevant document list, we filter the retrieval score by an appropriate threshold optimized on the training set and choose the top 10 documents if the filtered documents are more than 10.

Table 2
The evaluation results of our submission in BioASQ Task 12b Phase A - document retrieval using the system CPS-A.

	System	Mean Precision	Recall	F-Measure	MAP	GMAP
Batch 1	CPS-A	0.1119	0.0842	0.076	0.0661	0.0001
	Top Competitor	0.1294	0.3369	0.1728	0.2006	0.0019
Batch 2	CPS-A	0.0717	0.1201	0.073	0.0845	0.0001
	Top Competitor	0.1085	0.3580	0.1524	0.2041	0.0022
Batch 3	CPS-A	0.0576	0.1777	0.0778	0.116	0.0002
	Top Competitor	0.0980	0.3849	0.1438	0.2487	0.0027
Batch 4	CPS-A	0.0606	0.2304	0.0854	0.1352	0.0003
	Top Competitor	0.1239	0.5529	0.1833	0.3773	0.0142

4.2.2. Phase B

In this section, we demonstrate the 'ideal answer' submission results in BioASQ task 12b phase B using CPS-B in Table 3. Since there is randomness in answer generation, we submitted two or three results in the challenge and we recorded the best one in the table.

Table 3
The evaluation results of our submission in BioASQ Task 12b Phase B - ideal answer using the system CPS-B.

	System	R-2 (Rec)	R-2 (F1)	R-SU4 (Rec)	R-SU4 (F1)
Batch 1	CPS-B	0.2873	0.209	0.2936	0.2017
	Top Competitor	0.3288	0.3179	0.3353	0.3168
Batch 2	CPS-B	0.2413	0.1709	0.256	0.1694
	Top Competitor	0.4333	0.4008	0.4462	0.4035
Batch 3	CPS-B	0.3266	0.2544	0.3124	0.2394
	Top Competitor	0.5283	0.3806	0.5525	0.3643
Batch 4	CPS-B	0.2982	0.2147	0.3125	0.216
	Top Competitor	0.4398	0.3697	0.4208	0.3458

4.2.3. Phase A+

We used CPS-A+ to complete the submission of Phase A+, and the 'ideal answer' results are illustrated in Table 4. We did not submit the result of batch 1 due to the time limit.

Table 4
The evaluation results of our submission in BioASQ Task 12b Phase A+ - ideal answer using the system CPS-A+.

	System	R-2 (Rec)	R-2 (F1)	R-SU4 (Rec)	R-SU4 (F1)
Batch 2	CPS-A+	0.1729	0.1235	0.1949	0.1272
	Top Competitor	0.2187	0.1898	0.2292	0.1898
Batch 3	CPS-A+	0.2414	0.1955	0.2452	0.1946
	Top Competitor	0.3138	0.1454	0.3334	0.1470
Batch 4	CPS-A+	0.2224	0.1376	0.2517	0.1495
	Top Competitor	0.2783	0.2461	0.2795	0.2431

4.3. Ablation Study

4.3.1. On Generation Models

The capability of the answer generation model has a significant influence on the performance in Phase B. Here we compare the performance of different generation models when taking questions and relevant contexts as input (shown in Table 5).

The experiments in the first three rows of Table 5 were carried out by ourselves. In particular, the Llama2-7B-chat + LoRA system is identical to the mentioned system CPS-B, but its dataset has been changed as 4.1.1 stated. The difference between Llama2-7B-chat and CPS-B is that the fine-tuned model is replaced by the original Llama2-7B-chat model. The GPT-3.5-turbo system is based on the API of GPT-3.5-turbo, where the input consists of the query question and the context obtained from the retrieval process of the CPS-B system. In this way, the input in the case of GPT-3.5-turbo is consistent with that of CPS-B.

Apart from these systems, we also select baselines and outstanding results in BioASQ task 11b phase B[10]. BioASQ Baseline FS is an official baseline obtained from BioGPT [44], and is detailed described in [38]. This official baseline system uses the concatenation of the question body and the relevant snippets until the input length is exceeded. UR-gpt4 [39] and UR-gpt3.5 [40] are systems using ChatGPT4 and ChatGPT3.5 with corresponding snippets. IISR [41] also used ChatGPT as its main model, but used a different snippet input method as desctibed in 2.2.2.

We can observe that the GPT-4 model is quite powerful and can obtain impressive results without fine-tuning (the highest R-2 Recall and the second highest R-SU4 Recall). Besides, our system won the highest place among the three metrics including R-2 F1-score, R-SU4 Recall, and R-SU4 F1-score. Therefore, fine-tuning a relatively small LLM can achieve amazing results and beat a very powerful LLM in specific domains.

4.3.2. On Retrieved Context

Beyond the generation model itself, the retrieved context can improve the accuracy of the generated answers. Here we evaluated the performance while there is or isn't context given to the models, which is shown in Table 6. We can observe that all models didn't perform very well without relevant context. Llama2-7B-chat + LoRA and BioASQ Baseline ZS are better since they were fine-tuned on biomedical corpora. However, when the relevant context is given, all models have a significant improvement even more than 100% on some metrics. Therefore, Retrieval-Augmented Generation is useful and promising when dealing with domain-specific Qustion-Answering tasks.

[10] http://participants-area.bioasq.org/results/11b/phaseB/

Table 5
Performance of different generation models on BioASQ Task 11b - Phase B Test batch1

Model	R-2 (Rec)	R-2 (F1)	R-SU4 (Rec)	R-SU4 (F1)
GPT-3.5-turbo	0.2452	0.2999	0.2699	0.3258
Llama2-7B-chat	0.1599	0.1911	0.1181	0.1415
Llama2-7B-chat + LoRA (**CPS-B**)	0.5609	**0.5581**	**0.5798**	**0.5734**
BioASQ Baseline FS[38]	0.3048	0.2493	0.3026	0.2443
UR-gpt4 [39]	**0.5630**	0.2136	0.5521	0.1990
UR-gpt3.5 [40]	0.5245	0.1762	0.5209	0.1663
IISR [41]	0.4249	0.4037	0.4138	0.3930

Besides, we explored the influence of the retrieved context quality on the performance from two aspects: **Retrieval Unit** and **Retrieval Source**.

- **Retrieval Unit** The golden-enriched snippets provided in Phase B test sets are of course the best retrieval unit for the answer generation since they are validated by biomedical experts. Therefore, we build a baseline treating chunks split from the golden-enriched documents as retrieval units. The ensemble retriever needs to search the most relevant snippets or chunks to provide references for the answer generation model (Llama2-7B-chat + LoRA). The results are illustrated in Table 7, which indicates retrieving from the golden-enriched snippets brings better performance.
- **Retrieval Source** Then we experiment with adding noise to the retrieval process by changing the retrieval sources. The retrieval source means where the retrieval context comes from. We conducted experiments in three different settings: 'Test Set', 'Training Set', and 'Training + Test Set'. The 'Test Set' setting is identical to the setting in Phase B. The 'Training + Test Set' setting means the relevant snippets should be first searched from snippets in the training and test set, which will bring some noise to the given snippets. The 'Training Set' setting means the relevant snippets should be searched from snippets only in the training set. Notice that snippets in the training set could be irrelevant in most cases, so it will bring a huge amount of noise. The results in Table 8 also support the conclusion that higher quality of the retrieved contexts will lead to better performance.

Table 6
Comparison between systems using different answer generation models on the BioASQ Task 11b - Phase B Test batch1 dataset.

Model	Context	R-2 (Rec)	R-2 (F1)	R-SU4 (Rec)	R-SU4 (F1)
Llama2-7B-chat	w/o	0.0732	0.0951	0.0567	0.0713
Llama2-7B-chat	w/	0.1599	0.1911	0.1181	0.1415
Llama2-7B-chat + LoRA	w/o	0.2836	0.2481	0.2177	0.1814
Llama2-7B-chat + LoRA (**CPS-B**)	w/	**0.5609**	**0.5581**	**0.5798**	**0.5734**
GPT-3.5-turbo	w/o	0.0738	0.1089	0.093	0.1362
GPT-3.5-turbo	w/	0.2452	0.2999	0.2699	0.3258
BioASQ Baseline ZS[38]	w/o	0.1727	0.0977	0.1936	0.1004
BioASQ Baseline FS[38]	w/	0.3048	0.2493	0.3026	0.2443

4.3.3. On Context Input Method

When the generation model and the context were the same, we further explored different context input methods. We constructed a Baseline called 'Stuff Documents' here by replacing the ensemble retriever of the system CPS-B as the 'create stuff chain'[11] from the LangChain Package. This chain means stuffing

[11] Langchain API: langchain.chains.combine_documents.stuff.create_stuff_documents_chain

Table 7
Comparison between systems using different retrieval units on the BioASQ Task 11b - Phase B Test batch1 dataset.

Retrieval Unit	R-2 (Rec)	R-2 (F1)	R-SU4 (Rec)	R-SU4 (F1)
Snippet	**0.5609**	**0.5581**	**0.5798**	**0.5734**
Chunk	0.3616	0.3454	0.4042	0.3635

Table 8
Comparison between systems using different retrieval sources on the BioASQ Task 11b - Phase B Test batch1 dataset.

Retrieval Unit	Retrieval Source	R-2 (Rec)	R-2 (F1)	R-SU4 (Rec)	R-SU4 (F1)
Snippet	**Test Set**	**0.5609**	**0.5581**	**0.5798**	**0.5734**
Snippet	Training + Test set	0.4616	0.4793	0.4785	0.4934
Snippet	Train Set	0.2095	0.1917	0.2502	0.2235

all the relevant snippets of query questions as the context for the model input. The results in Table 9 show that introducing the retrieval process in CPS-B is more effective than just inputting all the relevant snippets as the context.

Table 9
Comparison between our retrieval-based CPS-B system and the 'Stuff Documents' baseline on the BioASQ Task 11b - Phase B Test batch1 dataset.

Method	R-2 (Rec)	R-2 (F1)	R-SU4 (Rec)	R-SU4 (F1)
CPS-B	**0.5609**	**0.5581**	**0.5798**	**0.5734**
Stuff Documents	0.3661	0.3183	0.4996	0.3572

5. Discussion

We constructed the **C**orpus **P**EFT **S**earching (**CPS**) system step by step following different phases in the BioASQ challenge. Firstly, in phase A the system acquires access to the PubMed Central database with a BM25 retriever. Secondly, in phase B the system integrates the Llama2-7b-chat model fine-tuned by LoRA with an ensemble retriever. Finally, in phase A+, we used an optimized two-stage hierarchical retrieval structure to connect the corpus and answer generation LLM. The first stage employs BM25 for a rough retrieval of documents, while the second stage combines BM25 with vector similarity for a fine-grained retrieval of document chunks. The CPS system can search documents related to query questions and generate appropriate answers. It achieved a mid to slightly higher position on the BioASQ 12b leaderboard rankings. The reason may be that the scale and the context window length of our model are relatively small.

However, we made great efforts to improve its performance by utilizing different techniques including sparse and dense ensemble retrievers, Parameter-Efficient Fine-Tuning (PEFT), and hierarchical Retrieval-Augmented Generation (RAG). The experiments in 4.3 demonstrate our exploration of the model fine-tuning and retrieval techniques. We could conclude that the techniques we employed could significantly enhance the model performance without altering its architecture, leveraging the full synthesis capabilities of a 7B-parameter scale LLM. The experiment and analysis elaborate the idea that if the retrieved context contains all the necessary information to complete the task, a small-scale LLM fine-tuned by a parameter-efficient method is fully capable of generating ideal answers. Besides, improving the context input strategy like using ensemble retrievers also has the potential to enhance system efficiency.

Experiment results illustrated that our method exceeded the best level of last year's ranking. However, there is a gap between the system performance in last year and this year's challenges. In this work, we only applied the BM25 method to retrieve documents or snippets, since BM25 is still a strong retriever in the biomedical RAG domain, according to the comparisons in Xiong et al. [33]. Dense retrievers built for the biomedical area such as MedCPT[32] have the potential to obtain better retrieval results. Future endeavors will focus on in-depth research in exploring more powerful retrieval techniques to improve the generation performance. In addition, the research on LLM has been varying. We will keep up with the pace and conduct more in-depth research and improvement on the generation model itself.

Acknowledgments

This work was supported by the IdeaBooster Fund IDBF23ENG05 of The Chinese University of Hong Kong.

References

[1] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al., Gpt-4 technical report, arXiv preprint arXiv:2303.08774 (2023).

[2] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).

[3] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al., Llama: Open and efficient foundation language models, arXiv preprint arXiv:2302.13971 (2023).

[4] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al., Llama 2: Open foundation and fine-tuned chat models, arXiv preprint arXiv:2307.09288 (2023).

[5] OpenAI, Introducing chatgpt, 2023. URL: https://openai.com/blog/chatgpt, openAI Blog, OpenAI.

[6] K. Singhal, T. Tu, J. Gottweis, R. Sayres, E. Wulczyn, L. Hou, K. Clark, S. Pfohl, H. Cole-Lewis, D. Neal, et al., Towards expert-level medical question answering with large language models, arXiv preprint arXiv:2305.09617 (2023).

[7] S. Wang, Z. Zhao, X. Ouyang, Q. Wang, D. Shen, Chatcad: Interactive computer-aided diagnosis on medical image using large language models, arXiv preprint arXiv:2302.07257 (2023).

[8] P. Cramer, Alphafold2 and the future of structural biology, Nature structural & molecular biology 28 (2021) 704–705.

[9] J. Chen, Z. Hu, S. Sun, Q. Tan, Y. Wang, Q. Yu, L. Zong, L. Hong, J. Xiao, T. Shen, et al., Interpretable rna foundation model from unannotated data for highly accurate rna structure and function predictions, arXiv preprint arXiv:2204.00300 (2022).

[10] Q. Yu, Z. Dong, X. Fan, L. Zong, Y. Li, Hmd-amp: Protein language-powered hierarchical multi-label deep forest for annotating antimicrobial peptides, arXiv preprint arXiv:2111.06023 (2021).

[11] M. Agrawal, S. Hegselmann, H. Lang, Y. Kim, D. Sontag, Large language models are few-shot clinical information extractors, arXiv preprint arXiv:2205.12689 (2022).

[12] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, et al., Retrieval-augmented generation for knowledge-intensive nlp tasks, Advances in Neural Information Processing Systems 33 (2020) 9459–9474.

[13] I. Klerings, A. S. Weinhandl, K. J. Thaler, Information overload in healthcare: too much of a good thing?, Zeitschrift für Evidenz, Fortbildung und Qualität im Gesundheitswesen 109 (2015) 285–290.

[14] A. Nentidis, G. Katsimpras, A. Krithara, S. Lima-López, E. Farré-Maduell, M. Krallinger, N. Loukachevitch, V. Davydova, E. Tutubalina, G. Paliouras, Overview of BioASQ 2024: The twelfth BioASQ challenge on Large-Scale Biomedical Semantic Indexing and Question Answering, in: L. Goeuriot, P. Mulhem, G. Quénot, D. Schwab, L. Soulier, G. Maria Di Nunzio, P. Galuščáková, A. García Seco de Herrera, G. Faggioli, N. Ferro (Eds.), Experimental IR Meets Multilinguality,

Multimodality, and Interaction. Proceedings of the Fifteenth International Conference of the CLEF Association (CLEF 2024), 2024.

[15] A. Nentidis, G. Katsimpras, A. Krithara, G. Paliouras, Overview of BioASQ Tasks 12b and Synergy12 in CLEF2024, in: G. Faggioli, N. Ferro, P. Galuščáková, A. García Seco de Herrera (Eds.), Working Notes of CLEF 2024 - Conference and Labs of the Evaluation Forum, 2024.

[16] A. Krithara, A. Nentidis, K. Bougiatiotis, G. Paliouras, BioASQ-QA: A manually curated corpus for Biomedical Question Answering, Scientific Data 10 (2023) 170.

[17] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, Y. Li, X. Wang, M. Dehghani, S. Brahma, et al., Scaling instruction-finetuned language models, Journal of Machine Learning Research 25 (2024) 1–53.

[18] D. Jin, E. Pan, N. Oufattole, W.-H. Weng, H. Fang, P. Szolovits, What disease does this patient have? a large-scale open domain question answering dataset from medical exams, Applied Sciences 11 (2021) 6421.

[19] A. Venigalla, J. Frankle, M. Carbin, Biomedlm: a domain-specific large language model for biomedical text, MosaicML. Accessed: Dec 23 (2022) 2.

[20] H. Xiong, S. Wang, Y. Zhu, Z. Zhao, Y. Liu, L. Huang, Q. Wang, D. Shen, Doctorglm: Fine-tuning your chinese doctor is not a herculean task, arXiv preprint arXiv:2304.01097 (2023).

[21] Y. Tan, M. Li, Z. Huang, H. Yu, G. Fan, Medchatzh: a better medical adviser learns from better instructions, arXiv preprint arXiv:2309.01114 (2023).

[22] T. Han, L. C. Adams, J.-M. Papaioannou, P. Grundmann, T. Oberhauser, A. Löser, D. Truhn, K. K. Bressem, Medalpaca–an open-source collection of medical conversational ai models and training data, arXiv preprint arXiv:2304.08247 (2023).

[23] H. Wang, C. Liu, N. Xi, Z. Qiang, S. Zhao, B. Qin, T. Liu, Huatuo: Tuning llama model with chinese medical knowledge, arXiv preprint arXiv:2304.06975 (2023).

[24] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, S. Gelly, Parameter-efficient transfer learning for nlp, in: International conference on machine learning, PMLR, 2019, pp. 2790–2799.

[25] X. L. Li, P. Liang, Prefix-tuning: Optimizing continuous prompts for generation, arXiv preprint arXiv:2101.00190 (2021).

[26] S. Robertson, H. Zaragoza, et al., The probabilistic relevance framework: Bm25 and beyond, Foundations and Trends® in Information Retrieval 3 (2009) 333–389.

[27] V. Karpukhin, B. Oğuz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, W.-t. Yih, Dense passage retrieval for open-domain question answering, arXiv preprint arXiv:2004.04906 (2020).

[28] P. Zhang, S. Xiao, Z. Liu, Z. Dou, J.-Y. Nie, Retrieve anything to augment large language models, arXiv preprint arXiv:2310.07554 (2023).

[29] M. Yasunaga, J. Leskovec, P. Liang, Linkbert: Pretraining language models with document links, arXiv preprint arXiv:2203.15827 (2022).

[30] M. Yasunaga, A. Bosselut, H. Ren, X. Zhang, C. D. Manning, P. S. Liang, J. Leskovec, Deep bidirectional language-knowledge graph pretraining, Advances in Neural Information Processing Systems 35 (2022) 37309–37323.

[31] X. Zhang, A. Bosselut, M. Yasunaga, H. Ren, P. Liang, C. D. Manning, J. Leskovec, Greaselm: Graph reasoning enhanced language models for question answering, arXiv preprint arXiv:2201.08860 (2022).

[32] Q. Jin, W. Kim, Q. Chen, D. C. Comeau, L. Yeganova, W. J. Wilbur, Z. Lu, Medcpt: Contrastive pre-trained transformers with large-scale pubmed search logs for zero-shot biomedical information retrieval, Bioinformatics 39 (2023) btad651.

[33] G. Xiong, Q. Jin, Z. Lu, A. Zhang, Benchmarking retrieval-augmented generation for medicine, arXiv preprint arXiv:2402.13178 (2024).

[34] Q. Jin, Y. Yang, Q. Chen, Z. Lu, Genegpt: Augmenting large language models with domain tools for improved access to biomedical information, Bioinformatics 40 (2024) btae075.

[35] M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. d. O. Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman, et al., Evaluating large language models trained on code, arXiv preprint

arXiv:2107.03374 (2021).

[36] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, et al., Chain-of-thought prompting elicits reasoning in large language models, Advances in neural information processing systems 35 (2022) 24824–24837.

[37] W. Shi, Y. Zhuang, Y. Zhu, H. Iwinski, M. Wattenbarger, M. D. Wang, Retrieval-augmented large language models for adolescent idiopathic scoliosis patients in shared decision-making, in: Proceedings of the 14th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics, 2023, pp. 1–10.

[38] A. Nentidis, G. Katsimpras, A. Krithara, S. Lima López, E. Farré-Maduell, L. Gasco, M. Krallinger, G. Paliouras, Overview of bioasq 2023: The eleventh bioasq challenge on large-scale biomedical semantic indexing and question answering, in: International Conference of the Cross-Language Evaluation Forum for European Languages, Springer, 2023, pp. 227–250.

[39] S. Ateia, U. Kruschwitz, Is chatgpt a biomedical expert, Exploring the Zero-Shot Performance of Current GPT Models in Biomedical Tasks (2023).

[40] A. Aksenova, T. Asamov, P. Ivanov, S. Boytcheva, Improving biomedical question answering with sentence-based ranking at bioasq-11b, in: Conference and Labs of the Evaluation Forum, 2023. URL: https://api.semanticscholar.org/CorpusID:264441330.

[41] C.-Y. Hsueh, Y. Zhang, Y.-W. Lu, J.-C. Han, W. Meesawad, R. T.-H. Tsai, Ncu-iisr: Prompt engineering on gpt-4 to stove biological problems in bioasq 11b phase b, in: 11th BioASQ Workshop at the 14th Conference and Labs of the Evaluation Forum (CLEF), 2023.

[42] J. Lin, X. Ma, S.-C. Lin, J.-H. Yang, R. Pradeep, R. Nogueira, Pyserini: A Python toolkit for reproducible information retrieval research with sparse and dense representations, in: Proceedings of the 44th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021), 2021, pp. 2356–2362.

[43] E. J. Hu, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, et al., Lora: Low-rank adaptation of large language models, in: International Conference on Learning Representations, 2021.

[44] R. Luo, L. Sun, Y. Xia, T. Qin, S. Zhang, H. Poon, T.-Y. Liu, Biogpt: generative pre-trained transformer for biomedical text generation and mining, Briefings in bioinformatics 23 (2022) bbac409.