Applying DeepSeek to BioASQ Task 13B: Using Supervised Fine-Tuning and Few-Shot Learning

Notebook for the BioASQ Lab at CLEF 2025

Jie Tang[1,2], Hua Yang[1,*], Kai Xiong[1,2], Hanyang Li[1,2], Paulo Quaresma[3], Hongbin Yu[1], Wenbo Zhang[4] and Mingzhou Song[1]

[1]School of Artificial Intelligence, Zhongyuan University of Technology, Zhengzhou, 450007, Henan, China
[2]School of Computer Science, Zhongyuan University of Technology, Zhengzhou, 450007, Henan, China
[3]Department of Computer Science, VISTA Lab, Algoritmi Center, University of Évora, 7000-671, Évora, Portugal
[4]Aquinas International Academy, 3200 E Guasti Rd, Suite 100, Ontario, 91761, CA, USA

Abstract

The recent surge in popularity of DeepSeek has attracted significant attention, yet its practical performance in real-world applications remains largely unexplored. In this study, our team participated in BioASQ Task 13b, which focuses on biomedical information retrieval and question answering (QA). We evaluated the DeepSeek model using three different approaches: local deployment, API-based access, and supervised fine-tuning. Specifically, we investigated the model's performance in few-shot learning settings. Notably, in Phase A+, our system using the deepseek-r1:671b model combined with retrieval-augmented generation techniques ranked first among all 67 submitted runs on yes/no questions in Batch 4. In Phase B, systems using both the deepseek-r1:32b and deepseek-r1:671b models achieved top performance on yes/no questions in Batches 2 and 3. Additionally, the system using the deepseek-r1:32b model ranked first on list questions in Batch 1. Our results demonstrate the proposed method is effective in biomedical QA tasks and shows promising potential for future applications in the domain. The code is available at https://github.com/wuren519/bioasq-2025.

Keywords

Large Language Model, Few-Shot Learning, Supervised Fine-tuning, Biomedical Information Retrieval, Biomedical Question Answering

CEUR-WS.org/Vol-4038/paper_44.pdf

1. Introduction

Biomedical question answering (QA) is a challenging task due to the complexity and domain specificity of medical terminology, concepts, and evidence retrieval[1]. The BioASQ challenge has long served as a benchmark for developing robust biomedical QA systems[2]. In recent years, Large Language Models (LLMs) have shown promise across a wide range of tasks, including open-domain QA and information retrieval[3].

In this work, we explore the use of LLMs[4] to build a biomedical QA system for the BioASQ Task 13b challenge[5]. We design an end-to-end pipeline that relies primarily on LLMs for query understanding, document retrieval, reranking, and answer generation. We participated in BioASQ Task 13b and submitted system results for Phase A, Phase A+, and Phase B. We compared prompt-based strategies with LLMs trained via supervised fine-tuning (SFT) on previous BioASQ data. Our system demonstrates that LLMs, with or without SFT, can serve as core components for biomedical QA.

Following the introduction, we will focus on related work in Section 2, describe our method in Section 3, report our results in Section 4, and provide conclusions and future outlook in Section 5.

CLEF 2025 Working Notes, 9–12 September 2025, Madrid, Spain
*Corresponding author.
2024007005@zut.edu.cn (J. Tang); huayang@zut.edu.cn (H. Yang); xiongkai2024@163.com (K. Xiong); 2023107324@zut.edu.cn (H. Li); pq@uevora.pt (P. Quaresma); hongbinyu@zut.edu.cn (H. Yu); zwbyz262@outlook.com (W. Zhang); mingzhou@zut.edu.cn (M. Song)
© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

2. Related Work

2.1. Biomedical QA

Biomedical QA systems aim to provide concise answers to specialized biomedical queries. The BioASQ task includes expert-written English questions of four types: yes/no, factoid, list, and summary[6]. In BioASQ Task 13b, systems retrieve relevant PubMed articles and snippets and produce an "exact" answer and an "ideal" answer for each question. For yes/no questions, the exact answer is "yes" or "no"; for factoid and list questions, it is the named entity or list of entities that answer the question; for summary questions, no exact answer is given and only the ideal answer[7]. Importantly, BioASQ provides all synonyms of the gold answers – e.g. a gene or disease may have multiple names – so systems must handle these alternative forms. The ideal answers are paragraph-sized explanations, intended to augment the exact answer. Therefore, the evaluation of the systems consists of two aspects: first, the assessment of exact answers, including accuracy or F1 score for yes/no questions, MRR for factoid questions, and F-Measure for list questions; second, the evaluation of summary quality, which is typically conducted through manual assessment or using ROUGE metrics.

2.2. LLMs for biomedical/health QA

The emergence of LLMs such as GPT, PaLM, and LLaMA has significantly advanced the field of QA, including biomedical and health domains[8]. These models, trained on vast corpora of general and domain-specific text, exhibit strong capabilities in understanding natural language queries, retrieving relevant information, and generating fluent, contextually appropriate answers. In the biomedical domain, this is particularly valuable due to the dense and specialized nature of medical texts[9].

Recent studies have demonstrated the utility of LLMs for biomedical QA tasks[3]. Models such as BioGPT[10] and PubMedGPT, pre-trained on biomedical literature, have shown improved performance on tasks like document classification, named entity recognition, and QA. Other works have explored prompt-based approaches, leveraging instruction-tuned LLMs to answer biomedical questions without task-specific SFT[11]. Despite promising results, challenges remain, including handling domain-specific terminology, ensuring factual correctness, and retrieving up-to-date evidence from sources like PubMed.

2.3. LLMs Supervised Fine-tuning

SFT has become a widely used strategy to adapt general-purpose LLMs to specific domains and tasks[12]. By training on labeled examples, SFT enables LLMs to align better with target outputs, adhere to domain-specific answer styles, and improve factuality. In biomedical QA, where precise terminology and structured responses are often required, SFT is particularly valuable.

Recent work has shown that models such as BioMedLM[13], BioGPT, and domain-adapted versions of T5[14] or BERT[15] benefit significantly from SFT on biomedical corpora or QA datasets. These models outperform zero-shot LLMs in tasks requiring structured output, such as factoid or list-type answers in BioASQ.

Given the promising results achieved by prior work using LLMs in biomedical QA tasks, and the strong performance of the DeepSeek models across various benchmarks, we explore the application of DeepSeek to biomedical QA and further fine-tune it for domain-specific adaptation.

3. Methodology

3.1. Phase A

Inspired by the work of Samy Ateia and Udo Kruschwitz[16], in Phase A, we built a query expansion-driven multi-stage retrieval and reranking framework based on PubMed. The framework consists of four components: query expansion, PubMed retrieval, document filtering, and snippet reranking. The detailed process is illustrated in Figure 1.

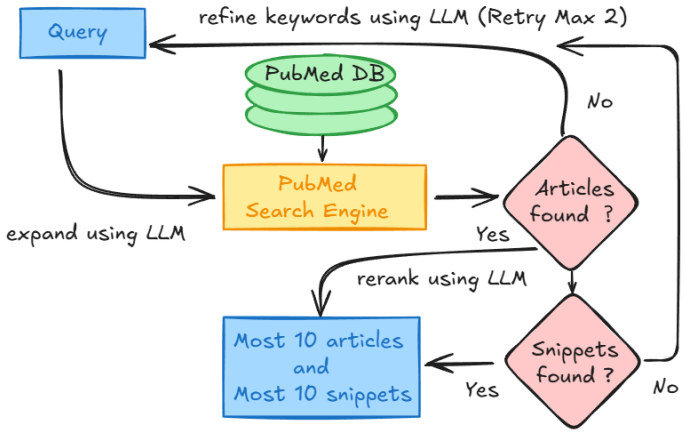

Figure 1: Flowchart of Phase A

We first perform few-shot query expansion on the original question. Using the DeepSeek model and a set of predefined examples, we generate a structured Boolean query expression. These predefined examples were selected by evaluating our own results on historical BioASQ datasets using F1 scores and choosing the highest-scoring entries, with the aim of enhancing the model's ability to handle current inputs by providing high-quality references. The output is then post-processed using regular expressions to remove redundant tags and adapt the format for the PubMed API.

Then, we use the PubMed API to retrieve articles based on the expanded query. The retrieval is limited to articles published between 2000 and 2025, in accordance with the BioASQ Task 13b requirement to use the 2025 PubMed baseline version. To ensure coverage of contemporary biomedical research, the year 2000 is selected as a reasonable starting point. If the initial query returns no results, a query refinement module is triggered. In this step, the original keywords that failed to retrieve articles are also included in the prompt, allowing LLMs to generate a broader query that retains the original context and relevance.

Next, we retrieve titles and abstracts for each PMID returned. We then use a large language model-based snippet extraction module to identify semantically relevant passages from the returned articles. If no relevant snippets are found, the system will re-trigger the query refinement and retrieval process, with a maximum of two iterations—a limit determined empirically: we tested 1, 2, and 3 iterations and found that two iterations yielded the best overall performance.

Finally, we rank the extracted snippets and reorder the original list of articles based on snippet relevance. This ensures that the most relevant snippets appear at the top of the final system output, enhancing answer usability and accuracy. The system ultimately selects the 10 most helpful snippets from the retrieved results.

3.2. Phase A+ and Phase B

The methods used in Phase A+ and Phase B are identical; the only difference lies in the source of the input documents and snippets. In Phase A+, the relevant articles and snippets are retrieved by our own system during Phase A, whereas in Phase B, they are officially provided and selected by experts. The detailed process is illustrated in Figure 2.

In both Phase A+ and Phase B, we adopt a question-type-specific prompting strategy to handle the four types of questions: yes/no, factoid, list, and summary. Table 1 illustrates the prompt specifically designed for list-type questions in Phase B, which resulted in the best performance of our model on the corresponding task. For each type, a tailored prompt is constructed, incorporating the relevant text snippets to help LLMs better answer the question. For yes/no, factoid, and list questions, the model is required to generate both the ‘exact_answer‘ and the "ideal_answer". For summary questions, only the

Table 1
Prompt and English Translation

Original Prompt (Chinese)	Prompt Translation (English)
[文献片段] [问题]：{问题} [要求]： 1. 返回严格符合此格式的JSON： {{"entities": ["实体1", "实体2"]}} 2. 实体必须来自文献片段 3. 按文献中出现频率排序（高频在前） 4. 每个实体必须是名词短语（2-5个单词） 5. 最多返回100个不同实体 6. 不要任何解释性文字 [示例]： 问题： What are the common symptoms of COVID-19? 响应：{{"entities": ["fever", "cough", "fatigue", "loss of smell"]}}	{Snippets} [Question]: {question} [Instructions]: 1. Return a JSON strictly in this format: {"entities": ["entity1", "entity2"]} 2. Entities must be extracted from the snippets 3. Sort entities by frequency of occurrence (high to low) 4. Each entity must be a noun phrase (2-5 words) 5. Return up to 100 unique entities 6. Do not include any explanatory text [Example]: Question: What are the common symptoms of COVID-19? Response: {{"entities": ["fever", "cough", "fatigue", "loss of smell"]}}

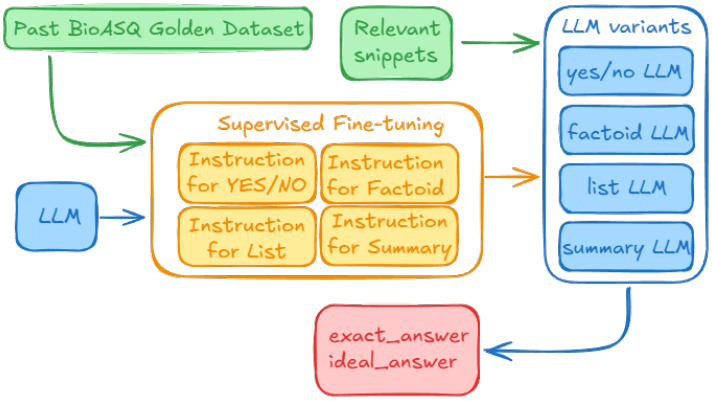

Figure 2: Flowchart of Phase A+ and Phase B

"ideal_answer" is needed.

Moreover, in our system, we apply SFT to a general-purpose open-source LLMs using historical BioASQ QA data. The model was trained to generate answers for all question types in Phase A+ and Phase B of BioASQ. In Phase A+,compared to the prompt-only version of our system, the SFT model shows improved consistency and factual accuracy, particularly for factoid and list questions.

4. Results and Analysis

4.1. Task 13 B Phase A

We participated in batches 1, 2, and 4 of Task 13b Phase A. Based on the work of Samy Ateia and Udo Kruschwitz [16], we limited the number of few-shot examples to 10 in all our systems. We used two models: deepseek-r1:32b, a 32-billion-parameter multimodal AI model developed by DeepSeek and

deployed locally, and deepseek-r1:671b, which was accessed via API.

Table 2[1] summarizes all the runs we submitted to the BioASQ 13b task, covering Phase A, A+, and B. Each row corresponds to a specific run configuration. The System Name column indicates the group of runs we designed, such as ZUT-IR-1 to ZUT-IR-3. Submit Name and Run denote the submission identifier and the order of the run within the system. Model refers to the large language model used in the run, while Original Name indicates the original name of the system. The last three columns (Phase A, Phase A+, and Phase B) specify whether the run was submitted to the respective phases. For example, ZUT-IR-2-b represents the second run under system ZUT-IR-2, which uses the DeepSeek-r1:8b-distill model. This run was submitted to both Phase A+ and Phase B, and its original name is deepseek32b-full.

Table 2
Implemented Runs

System Name	Submit Name	Run	Model	Original Name	Phase A	Phase A+	Phase B
ZUT-IR-1 (a–e)	-a	first run	deepseek-r1:32b	deepseek32b-me/deepseek32b-f	✓	✓	✓
	-b	second run		deepseek32b-full/phaseB-4	○	✓	✓
	-c	third run		deepseek32b-f	○	✓	✓
	-d	fourth run		phaseB-4	○	○	✓
	-e	fifth run		phaseB-5	○	○	✓
ZUT-IR-2 (a–b)	-a	first run	deepseek-r1:8b-distill	deepseek32b-me	○	✓	✓
	-b	second run		deepseek32b-full	○	✓	✓
ZUT-IR-3 (a–b)	-a	first run	deepseek-r1:671b	phaseB-4/deepseek32b-full	✓	✓	✓
	-b	second run		phaseB-5	○	✓	✓

Tables 3 and 4[2] present the complete results of our participation in the three batches of Task 13b Phase A, where MAP is the primary official evaluation metric.

Table 3
Task 13B Phase A, Document Retrieval

Batch	Position	System	Precision	Recall	F-Measure	MAP	GMAP
1	1 of 51	Top Competitor	0.1047	0.5043	0.1605	0.4246	0.0104
	35 of 51	ZUT-IR-3-a	0.1514	0.2298	0.1641	0.1712	0.0002
	36 of 51	ZUT-IR-1-a	0.1378	0.1690	0.1425	0.1428	0.0001
2	1 of 41	Top Competitor	0.0976	0.5093	0.1546	0.4425	0.0096
	33 of 41	ZUT-IR-3-a	0.1170	0.2131	0.1285	0.1738	0.0003
	34 of 41	ZUT-IR-1-a	0.0611	0.2059	0.0863	0.1492	0.0002
4	1 of 79	Top Competitor	0.0600	0.2512	0.0927	0.1801	0.0008
	23 of 79	ZUT-IR-1-a	0.0741	0.1595	0.0809	0.1014	0.0001
	43 of 79	ZUT-IR-3-a	0.0513	0.1047	0.0565	0.0586	0.0001

We compared the performance of deepseek-r1:671b and deepseek-r1:32b on Task 13b Phase A. In both batch 1 and batch 2, the system using deepseek-r1:671b outperformed the one using deepseek-r1:32b in terms of MAP evaluation, in both document retrieval and snippet extraction. In batch 4, the system using deepseek-r1:32b significantly outperformed the one using deepseek-r1:671b in terms of MAP evaluation, both in document retrieval and snippet extraction. Among them, for document retrieval, the MAP score of the deepseek-r1:32b-based system was 0.1014, whereas the score of the deepseek-r1:671b-based system was only 0.0586. Similar observations have been reported in previous studies. Srivastava et al. [17] found in the BIG-bench benchmark that different models exhibit significant variation in performance across tasks, which may be due to the alignment between

[1]Only in Phase B, the Original Names of ZUT-IR-1-a and ZUT-IR-1-b are deepseek32b-f and phaseB-4, respectively. In all other phases, their Original Names are deepseek32b-me and deepseek32b-full. Similarly, only in Phase A, the Original Name of ZUT-IR-3-a is deepseek32b-full, while in other phases it corresponds to phaseB-4. In the fourth batch of Phase A+, the Original Name of ZUT-IR-1-a is deepseek32b-f, while in the other batches, it is deepseek32b-me.

[2]"Top Competitor" refers to the highest-ranked system in that batch, which is not ours. When the top competitor is absent in a reported batch, one of our systems is the best-performing one.

Table 4
Task 13B Phase A, Snippet Extraction

Batch	Position	System	Precision	Recall	F-Measure	MAP	GMAP
1	1 of 51	Top Competitor	0.0803	0.3050	0.1186	0.4535	0.0014
	25 of 51	ZUT-IR-3-a	0.0998	0.0575	0.0651	0.1131	0.0001
	26 of 51	ZUT-IR-1-a	0.0955	0.0517	0.0603	0.1085	0.0001
2	1 of 41	Top Competitor	0.0941	0.3625	0.1421	0.5522	0.0035
	23 of 41	ZUT-IR-3-a	0.0812	0.1200	0.0780	0.1883	0.0002
	27 of 41	ZUT-IR-1-a	0.0884	0.0985	0.0690	0.1523	0.0002
4	1 of 79	Top Competitor	0.0411	0.1135	0.0560	0.1634	0.0001
	15 of 79	ZUT-IR-1-a	0.0488	0.0939	0.0548	0.0682	0.0001
	26 of 79	ZUT-IR-3-a	0.0428	0.0520	0.0368	0.0343	0.0000

task characteristics and model capabilities. Furthermore, Wei et al. [18] pointed out that in certain reasoning tasks, smaller models may outperform larger ones due to their preference for specific patterns. Therefore, our observation that deepseek-r1:32b outperforms larger models in Batch 4 may be attributed to a distribution of questions that better aligns with its strengths.

4.2. Task 13B Phase A+

In Phase A+, we also used 10-shot learning across all systems. The specific system configurations are shown in Table 2. The majority of our systems leveraged the deepseek-r1:32b model, with the deepseek-r1:671b model used selectively in later-phase systems. In batches 3 and 4, we further conducted SFT on the deepseek-r1:8b model as part of our experiments.

Table 5 presents the yes/no results for exact questions in Phase A+. In batch 1, for the exact yes/no questions, we submitted two systems, both based on the deepseek-r1:32b model. In batch 2, we submitted five systems. The results show that both systems using deepseek-r1:671b outperformed the three systems based on deepseek-r1:32b. Among them, the best-performing system was the second run based on the deepseek-r1:671b model, which ranked 10th out of all 49 submitted systems. In batch 3, we submitted two systems, both using the SFT deepseek-r1:8b model. The results indicated that both systems performed poorly. In batch 4, we submitted five systems. Among them, the "-a" systems utilized relevant snippets retrieved in Phase A by the deepseek-r1:32b model, while the "-b" systems used snippets obtained in Phase A by the deepseek-r1:671b model. Among the two systems using the deepseek-r1:671b model in Phase A+, ZUT-IR-3-a—whose snippets were retrieved by deepseek-r1:32b in Phase A—ranked 1st out of all 67 submitted systems; in contrast, ZUT-IR-3-b—whose snippets came from deepseek-r1:671b—ranked only 26th. A similar trend was observed between ZUT-IR-2-a and ZUT-IR-2-b, where the only difference was the snippet retrieval model used. These observations underscore the critical impact of snippet retrieval quality on answer generation performance in Phase A+. It is worth noting that the systems based on the SFT deepseek-r1:8b model performed very poorly in batch 3, and were also outperformed in batch 4 by systems using both deepseek-r1:671b and deepseek-r1:32b.

Table 6 presents the factoid results for exact questions in Phase A+. For the exact factoid questions, the system configurations across batches were consistent with those described for the exact yes/no questions. In batch 1, the two systems we submitted based on the deepseek-r1:32b model performed poorly. In batch 2, we submitted five systems. Consistent with the results for the exact yes/no questions, the two systems using deepseek-r1:671b outperformed the three systems based on deepseek-r1:32b. In batch 3, the two systems (ZUT-IR-2-a and ZUT-IR-2-b) based on the supervised fine-tuned deepseek-r1:8b model achieved competitive results, ranking 16th and 17th among the 58 submitted systems. In batch 4, results showed that the two systems using the SFT deepseek-r1:8b model achieved the best performance, followed by the systems based on deepseek-r1:32b, while the two systems using

Table 5
Task 13 B, Phase A+, exact questions Yes/No

Batch	Position	System	Accuracy	F1 Yes	F1 No	Macro F1
1	1 of 56	Top Competitor	1.0000	1.0000	1.0000	1.0000
	51 of 56	ZUT-IR-1-a	0.2941	—	0.4545	0.9377
	56 of 56	ZUT-IR-1-b	0.2941	—	0.4545	0.2273
2	1 of 49	Top Competitor	1.0000	1.0000	1.0000	1.0000
	10 of 49	ZUT-IR-3-b	0.9412	0.9524	0.9231	0.9377
	11 of 49	ZUT-IR-3-a	0.9412	0.9524	0.9231	0.9377
	18 of 49	ZUT-IR-1-c	0.8824	0.9091	0.8333	0.8712
	21 of 49	ZUT-IR-1-b	0.8824	0.9167	0.8000	0.7571
	27 of 49	ZUT-IR-1-a	0.8235	0.8800	0.6667	0.7733
3	1 of 58	Top Competitor	0.9545	0.9697	0.9091	0.9394
	45 of 58	ZUT-IR-2-a	0.7273	0.8000	0.5714	0.6857
	46 of 58	ZUT-IR-2-b	0.7273	0.8000	0.5714	0.6857
4	1 of 67	**ZUT-IR-3-a**	**0.9231**	**0.9474**	**0.8571**	**0.9023**
	26 of 67	ZUT-IR-3-b	0.8846	0.9189	0.8000	0.7658
	39 of 67	ZUT-IR-1-a	0.8462	0.8889	0.7500	0.8194
	41 of 67	ZUT-IR-2-a	0.8077	0.8718	0.6154	0.7436
	52 of 67	ZUT-IR-2-b	0.7692	0.8421	0.5714	0.7068

deepseek-r1:671b performed the worst. Among the two systems based on the SFT deepseek-r1:8b model, the one that utilized snippets retrieved by deepseek-r1:32b in Phase A outperformed the one that used snippets retrieved by deepseek-r1:671b, further confirming the critical impact of snippet retrieval quality on answer generation performance in Phase A+.

Table 6
Task 13 B, Phase A+, exact questions factoid

Batch	Position	System	Strict Acc.	Lenient Acc.	MRR
1	1 of 56	Top Competitor	0.4231	0.4615	0.4423
	42 of 56	ZUT-IR-1-a	0.0769	0.0769	0.0769
	46 of 56	ZUT-IR-1-b	0.0769	0.0769	0.0769
2	1 of 49	Top Competitor	0.5926	0.5926	0.5926
	24 of 49	ZUT-IR-3-b	0.3333	0.3333	0.3333
	25 of 49	ZUT-IR-3-a	0.3333	0.3333	0.3333
	30 of 49	ZUT-IR-1-a	0.2963	0.2963	0.2963
	31 of 49	ZUT-IR-1-b	0.2963	0.2963	0.2963
	33 of 49	ZUT-IR-1-c	0.2963	0.2963	0.2963
3	1 of 58	Top Competitor	0.3500	0.5926	0.5926
	16 of 58	ZUT-IR-2-a	0.2500	0.2500	0.2500
	17 of 58	ZUT-IR-2-b	0.2500	0.2500	0.2500
4	1 of 67	Top Competitor	0.5926	0.5926	0.5926
	10 of 67	ZUT-IR-2-a	0.4545	0.4545	0.4545
	24 of 67	ZUT-IR-2-b	0.4091	0.4091	0.4091
	39 of 67	ZUT-IR-1-a	0.3636	0.3636	0.3636
	40 of 67	ZUT-IR-3-b	0.3636	0.3636	0.3636
	41 of 67	ZUT-IR-3-a	0.3636	0.3636	0.3636

Table 7 presents the list results for exact questions in Phase A+. For the exact list questions, the system configurations across batches were also consistent with those described for the exact yes/no questions. In batch 1, the two systems we submitted using the deepseek-r1:32b model performed well. In batch 2, we submitted five systems. The results show that the three systems using deepseek-r1:32b outperformed both systems based on deepseek-r1:671b. In batch 3, the results indicated that the two

systems using the SFT deepseek-r1:8b model did not perform well and ranked relatively low. In batch 4, the system we submitted using the SFT deepseek-r1:8b model achieved the best result, ranking third among all 67 submitted systems. This system also relied on snippets retrieved by deepseek-r1:32b in Phase A.

Table 7
Task 13 B, Phase A+, exact questions list

Batch	Position	System	Mean Prec.	Recall	F-Measure
1	1 of 56	Top Competitor	0.2362	0.2370	0.2330
	10 of 56	ZUT-IR-1-a	0.2078	0.1751	0.1843
	14 of 56	ZUT-IR-1-b	0.1977	0.1773	0.1801
2	1 of 49	Top Competitor	0.3785	0.4357	0.3880
	9 of 49	ZUT-IR-1-b	0.3038	0.3072	0.2955
	18 of 49	ZUT-IR-1-c	0.2458	0.3160	0.2675
	19 of 49	ZUT-IR-3-b	0.2395	0.2984	0.2574
	20 of 49	ZUT-IR-3-a	0.2395	0.2984	0.2574
	22 of 49	ZUT-IR-1-a	0.2199	0.3116	0.2390
3	1 of 58	Top Competitor	0.4674	0.4446	0.4541
	43 of 58	ZUT-IR-2-b	0.1504	0.1742	0.1580
	46 of 58	ZUT-IR-2-a	0.0876	0.1250	0.1000
4	1 of 67	Top Competitor	0.3272	0.2573	0.2845
	3 of 67	**ZUT-IR-2-a**	**0.3217**	**0.2929**	**0.3014**
	8 of 67	ZUT-IR-2-b	0.2775	0.2673	0.2670
	17 of 67	ZUT-IR-3-a	0.2549	0.2911	0.2652
	28 of 67	ZUT-IR-3-b	0.2263	0.2244	0.2210
	39 of 67	ZUT-IR-1-a	0.2095	0.2543	0.2165

Notably, in Phase A+, our SFT deepseek-r1:8b model performed well on both the exact factoid and exact list questions, outperforming the locally deployed deepseek-r1:32b model and the API-based deepseek-r1:671b model. However, its performance on the exact yes/no questions was suboptimal. This performance pattern suggests that SFT may be particularly effective in guiding the model to produce well-structured answers for factoid and list-type questions, which often follow predictable formats. In contrast, yes/no questions typically require more nuanced semantic understanding and inferential reasoning, which may benefit from the broader knowledge capacity and emergent reasoning abilities of larger, non-fine-tuned models such as deepseek-r1:671b.

4.3. Task 13B Phase B

In Phase B, we also used 10-shot learning across all systems. The specific system configurations are shown in Table 2. We submitted five systems in each of the four batches. We used three types of models: a locally deployed deepseek-r1:32b model, an API-accessed deepseek-r1:671b model, and a SFT deepseek-r1:8b model.

Table 8 presents the yes/no results for exact questions in Phase B. In batch 1, for the exact yes/no questions, all five systems using the deepseek-r1:32b model performed poorly. In batch 2, the results showed that the two systems using the deepseek-r1:671b model outperformed the three systems based on deepseek-r1:32b. Moreover, these two deepseek-r1:671b systems tied for first place among all 72 submitted systems, achieving an accuracy of 1. In batch 3, our submitted system ZUT-IR-1-b and system ZUT-IR-3-a tied for first place among all 66 submitted systems. In batch 4, the best-performing system was ZUT-IR-3-b, which ranked 9th out of 79 submitted systems. The other four systems achieved exactly the same accuracy. It is worth noting that the deepseek-r1:671b model consistently outperformed its smaller counterparts in handling yes/no questions across multiple batches in Phase B. This observation suggests that larger models may possess stronger generalization and inferential reasoning capabilities, which are particularly beneficial for binary classification tasks. In contrast, the SFT deepseek-r1:8b model exhibited relatively poor and inconsistent performance, possibly due to limited coverage or

distributional bias in the fine-tuning data, which may hinder its ability to handle semantically diverse or ambiguous yes/no questions.

Table 8
Task 13 B, Phase B, exact questions Yes/No

Batch	Position	System	Accuracy	F1 Yes	F1 No	Macro F1
1	1 of 72	Top Competitor	1.0000	1.0000	1.0000	1.0000
	68 of 72	ZUT-IR-1-a	0.2941	0.0000	0.4545	0.2273
	69 of 72	ZUT-IR-1-b	0.2941	0.0000	0.4545	0.2273
	70 of 72	ZUT-IR-1-c	0.2941	0.0000	0.4545	0.2273
	71 of 72	ZUT-IR-1-d	0.2941	0.0000	0.4545	0.2273
	72 of 72	ZUT-IR-1-e	0.2941	0.0000	0.4545	0.2273
2	1 of 72	**ZUT-IR-3-a**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
	1 of 72	**ZUT-IR-3-b**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
	37 of 72	ZUT-IR-1-b	0.9412	0.9524	0.9231	0.9377
	38 of 72	ZUT-IR-1-c	0.9412	0.9524	0.9231	0.9377
	53 of 72	ZUT-IR-1-a	0.8824	0.9000	0.8571	0.8786
3	1 of 66	**ZUT-IR-1-b**	**0.9545**	**0.9697**	**0.9091**	**0.9394**
	1 of 66	**ZUT-IR-3-a**	**0.9545**	**0.9697**	**0.9091**	**0.9394**
	36 of 66	ZUT-IR-2-a	0.9091	0.9375	0.8333	0.8854
	37 of 66	ZUT-IR-2-b	0.9091	0.9375	0.8333	0.8854
	38 of 66	ZUT-IR-1-a	0.9091	0.9412	0.8000	0.8706
4	1 of 79	Top Competitor	1.0000	1.0000	1.0000	1.0000
	9 of 79	ZUT-IR-3-b	0.9615	0.9744	0.9231	0.9487
	32 of 79	ZUT-IR-2-a	0.9231	0.9474	0.8571	0.9023
	33 of 79	ZUT-IR-2-b	0.9231	0.9474	0.8571	0.9023
	34 of 79	ZUT-IR-1-a	0.9231	0.9474	0.8571	0.9023
	35 of 79	ZUT-IR-3-a	0.9231	0.9474	0.8571	0.9023

Table 9 presents the factoid results for exact questions in Phase B. In batch 1, for the exact factoid questions, all five of our systems used the deepseek-r1:32b model. The results indicated poor performance. In batch 2, the results showed that the three systems using the deepseek-r1:32b model outperformed the two systems based on deepseek-r1:671b. Our best result came from the second run using the deepseek-r1:32b model, which ranked 5th among all 72 submitted systems. In batch 3, our best-performing system also used the deepseek-r1:32b model, ranking 10th among all 66 systems — outperforming the systems that used the SFT deepseek-r1:8b model and the deepseek-r1:671b model. In batch 4, the best performance was achieved by the two systems that employed the deepseek-r1:671b model.

Table 10 presents the list results for exact questions in Phase B.In batch 1, the system run with the deepseek-r1:32b model for the fourth time achieved the best performance across the entire competition, ranking first among all 72 submitted systems. Additionally, the systems run with the deepseek-r1:32b model during the first and second runs ranked fourth and sixth, respectively. In batch 2, two systems utilizing the deepseek-r1:671b model outperformed three systems based on the deepseek-r1:32b model. The best-performing system in this batch was ZUT-IR-3-b, which ranked eighth among all 72 systems. In batch 3, our system ZUT-IR-1-b achieved the highest ranking, placing fifth among 66 systems. The results indicated that systems employing the SFT deepseek-r1:8b model exhibited the poorest overall performance. In batch 4, consistent with previous observations, systems using the deepseek-r1:671b model achieved the best results, whereas those based on the SFT deepseek-r1:8b model performed comparatively poorly.

5. Conclusion and Future Work

We demonstrated the advanced performance of the emerging DeepSeek models in biomedical retrieval scenarios when combined with 10-shot learning. In BioASQ Task 13b, we utilized three different configurations of DeepSeek models: the locally deployed deepseek-r1:32b, the SFT deepseek-r1:8b, and

Table 9
Task 13 B, Phase B, exact questions factoid

Batch	Position	System	Strict Acc.	Lenient Acc.	MRR
1	1 of 72	Top Competitor	0.5385	0.6538	0.5962
	46 of 72	ZUT-IR-1-a	0.3846	0.3846	0.3846
	48 of 72	ZUT-IR-1-b	0.3846	0.3846	0.3846
	49 of 72	ZUT-IR-1-c	0.2273	0.3846	0.3846
	50 of 72	ZUT-IR-1-e	0.2273	0.3846	0.3846
	55 of 72	ZUT-IR-1-d	0.3462	0.3462	0.3462
2	1 of 72	Top Competitor	0.7037	0.7037	0.7037
	5 of 72	ZUT-IR-1-b	0.6667	0.6667	0.6667
	9 of 72	ZUT-IR-1-c	0.5926	0.5926	0.5926
	18 of 72	ZUT-IR-1-a	0.5556	0.5556	0.5556
	29 of 72	ZUT-IR-3-a	0.5185	0.5185	0.5185
	51 of 72	ZUT-IR-3-b	0.4074	0.4074	0.4074
3	1 of 66	Top Competitor	0.4500	0.600	0.5042
	10 of 66	ZUT-IR-1-a	0.4000	0.4000	0.4000
	12 of 66	ZUT-IR-1-b	0.4000	0.4000	0.4000
	24 of 66	ZUT-IR-2-a	0.3500	0.3500	0.3500
	25 of 66	ZUT-IR-2-b	0.3500	0.3500	0.3500
	36 of 66	ZUT-IR-3-a	0.3000	0.3000	0.3000
4	1 of 72	Top Competitor	0.6364	0.6364	0.6364
	30 of 72	ZUT-IR-3-a	0.5000	0.5000	0.5000
	31 of 72	ZUT-IR-3-b	0.5000	0.5000	0.5000
	40 of 72	ZUT-IR-2-a	0.4545	0.4545	0.4545
	41 of 72	ZUT-IR-2-b	0.4545	0.4545	0.4545
	42 of 72	ZUT-IR-1-a	0.4545	0.4545	0.4545

Table 10
Task 13 B, Phase B, exact questions list

Batch	Position	System	Mean Prec.	Recall	F-Measure
1	1 of 72	**ZUT-IR-1-d**	0.6226	0.5588	0.5808
	4 of 72	ZUT-IR-1-a	0.6022	0.6022	0.5769
	6 of 72	ZUT-IR-1-b	0.5826	0.5639	0.5659
	19 of 72	ZUT-IR-1-e	0.5493	0.5047	0.5158
	37 of 72	ZUT-IR-1-c	0.4770	0.4385	0.4507
2	1 of 72	Top Competitor	0.6842	0.2308	0.3329
	8 of 72	ZUT-IR-3-b	0.5705	0.5748	0.5412
	19 of 72	ZUT-IR-3-a	0.5410	0.6296	0.5408
	28 of 72	ZUT-IR-1-a	0.5079	0.5770	0.5182
	38 of 72	ZUT-IR-1-c	0.4744	0.5336	0.4846
	39 of 72	ZUT-IR-1-b	0.4718	0.5073	0.4723
3	1 of 66	Top Competitor	0.6659	0.6530	0.6331
	5 of 66	ZUT-IR-1-b	0.6417	0.5999	0.6102
	13 of 66	ZUT-IR-1-a	0.6247	0.6050	0.5976
	24 of 66	ZUT-IR-3-a	0.5770	0.5494	0.5502
	29 of 66	ZUT-IR-2-a	0.5433	0.5631	0.5473
	30 of 66	ZUT-IR-2-b	0.5433	0.5631	0.5473
4	1 of 79	Top Competitor	0.7491	0.5980	0.6492
	23 of 79	ZUT-IR-3-b	0.5558	0.6055	0.5648
	32 of 79	ZUT-IR-1-a	0.5098	0.5156	0.5043
	37 of 79	ZUT-IR-3-a	0.4921	0.4985	0.4871
	53 of 79	ZUT-IR-2-a	0.3994	0.4293	0.3987
	54 of 79	ZUT-IR-2-b	0.3994	0.4293	0.3987

the API-based deepseek-r1:671b. Notably, in Phase A+, our system using the deepseek-r1:671b model combined with retrieval-augmented generation techniques ranked first among all 67 submitted runs on yes/no questions in Batch 4. In Phase B, systems using both the deepseek-r1:32b and deepseek-r1:671b

models achieved top performance on yes/no questions in Batches 2 and 3. Additionally, the system using the deepseek-r1:32b model ranked first on list questions in Batch 1. Our SFT deepseek-r1:8b model performed well only on the exact factoid and exact list questions in Phase A+, but showed poor performance in other phases.

Our findings suggest that SFT may be particularly effective in guiding the model to produce well-structured answers for factoid and list-type questions, which often follow predictable formats. In contrast, yes/no questions typically require more nuanced semantic understanding and inferential reasoning, which may benefit more from the broader knowledge capacity and emergent reasoning abilities of larger, non-fine-tuned models such as deepseek-r1:671b.

In the future, we will conduct in-depth research on the application of fine-tuned models in Task 13b, aiming to further explore their capabilities and limitations across different question types and retrieval settings.

Acknowledgments

This research work was funded by:1) the Key Scientific Research Project of Higher Education Institutions in Henan Province, grant no. 24A520060. 2) Graduate Education and Teaching Reform Research Project of Zhongyuan University of Technology, grant no. JG202434.

Declaration on Generative AI

During the preparation of this manuscript, the author used ChatGPT for text translation. After utilizing this tool, the author carefully reviewed, revised, and edited the translated content, and assumes full responsibility for the accuracy and integrity of the final version.

References

[1] B.-C. Chih, J.-C. Han, R. Tzong-Han Tsai, Ncu-iisr: enhancing biomedical question answering with gpt-4 and retrieval augmented generation in bioasq 12b phase b, CLEF Working Notes (2024).

[2] A. Nentidis, G. Katsimpras, A. Krithara, M. Krallinger, M. Rodríguez-Ortega, E. Rodriguez-López, N. Loukachevitch, A. Sakhovskiy, E. Tutubalina, D. Dimitriadis, G. Tsoumakas, G. Giannakoulas, A. Bekiaridou, A. Samaras, G. M. D. Nunzio, N. Ferro, S. Marchesin, M. Martinelli, G. Silvello, G. Paliouras, Overview of bioasq 2025: The thirteenth bioasq challenge on large-scale biomedical semantic indexing and question answering, in: J. C. de Albornoz, J. Gonzalo, L. Plaza, A. G. S. de Herrera, J. Mothe, F. Piroi, P. Rosso, D. Spina, G. Faggioli, N. Ferro (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Sixteenth International Conference of the CLEF Association (CLEF 2025), 2025.

[3] W. Zhou, T. H. Ngo, Using pretrained large language model with prompt engineering to answer biomedical questions, arXiv preprint arXiv:2407.06779 (2024).

[4] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, Advances in neural information processing systems 33 (2020) 1877–1901.

[5] A. Nentidis, G. Katsimpras, A. Krithara, G. Paliouras, Overview of BioASQ Tasks 13b and Synergy13 in CLEF2025, in: G. Faggioli, N. Ferro, P. Rosso, D. Spina (Eds.), CLEF 2025 Working Notes, 2025.

[6] A. Krithara, A. Nentidis, K. Bougiatiotis, G. Paliouras, Bioasq-qa: A manually curated corpus for biomedical question answering, Scientific Data 10 (2023) 170.

[7] Q. Jin, Z. Yuan, G. Xiong, Q. Yu, H. Ying, C. Tan, M. Chen, S. Huang, X. Liu, S. Yu, Biomedical question answering: a survey of approaches and challenges, ACM Computing Surveys (CSUR) 55 (2022) 1–36.

[8] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, G. Lample, Llama: Open and efficient foundation language models, 2023. URL: https://arxiv.org/abs/2302.13971. arXiv:2302.13971.

[9] Q. Chen, Y. Hu, X. Peng, Q. Xie, Q. Jin, A. Gilson, M. B. Singer, X. Ai, P.-T. Lai, Z. Wang, et al., Benchmarking large language models for biomedical natural language processing applications and recommendations, Nature communications 16 (2025) 3280.

[10] 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.

[11] J. Wei, M. Bosma, V. Y. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, Q. V. Le, Finetuned language models are zero-shot learners, arXiv preprint arXiv:2109.01652 (2021).

[12] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, et al., Training language models to follow instructions with human feedback, Advances in neural information processing systems 35 (2022) 27730–27744.

[13] E. Bolton, A. Venigalla, M. Yasunaga, D. Hall, B. Xiong, T. Lee, R. Daneshjou, J. Frankle, P. Liang, M. Carbin, et al., Biomedlm: A 2.7 b parameter language model trained on biomedical text, arXiv preprint arXiv:2403.18421 (2024).

[14] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, Journal of machine learning research 21 (2020) 1–67.

[15] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, 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: human language technologies, volume 1 (long and short papers), 2019, pp. 4171–4186.

[16] S. Ateia, U. Kruschwitz, Can open-source llms compete with commercial models? exploring the few-shot performance of current gpt models in biomedical tasks, arXiv preprint arXiv:2407.13511 (2024).

[17] A. Srivastava, A. Rastogi, A. Rao, A. A. M. Shoeb, A. Abid, A. Fisch, A. R. Brown, A. Santoro, A. Gupta, A. Garriga-Alonso, et al., Beyond the imitation game: Quantifying and extrapolating the capabilities of language models, arXiv preprint arXiv:2206.04615 (2022).

[18] J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler, et al., Emergent abilities of large language models, arXiv preprint arXiv:2206.07682 (2022).