NCU-IISR: A Retrieval-Augmented Generation Approach for BioASQ 13b Phase A and A+

Notebook for the BioASQ Lab at CLEF 2025

Jen-Chieh Han[a], Bing-Chen Chih[a], Hsi-Chuan Hung[b] and Richard Tzong-Han Tsai[a,b,c,*]

[a] *Department of Computer Science and Information Engineering, National Central University, Taoyuan, Taiwan*
[b] *Department of Medical Research, Cathay General Hospital, Taipei, Taiwan*
[c] *Center for GIS, Research Center for Humanities and Social Sciences, Academia Sinica, Taipei, Taiwan*

Abstract

In this study, we employed a basic Retrieval-Augmented Generation (RAG) framework to sequentially address the BioASQ 13b Phase A and A+ tasks. Our pipeline consists of three main components: a retriever, a reranker, and a large language model (LLM) for natural language generation (NLG). We used the BM25 algorithm to retrieve candidate documents from the PubMed 2024 corpus, which included article titles and abstracts. The initially retrieved candidate documents were further re-ranked using the BAAI/bge-reranker-v2-m3 model to identify the most relevant articles and, after sentence segmentation, the most relevant snippets. For answer generation, we employed both the meta-llama/Llama-3.1-8B-Instruct model and GPT-4o. Furthermore, for the Phase A+ task, we extended the answer generation pipeline previously developed by Chih et al. for Phase B, allowing for a comparative evaluation between two distinct generation strategies.

Keywords

Biomedical Question Answering, Retrieval-Augmented Generation, BM25, BGE Reranker, LLMs, BioASQ

1. Introduction

The rapid advancement of large language models (LLMs) has led to the explosive success of general-purpose question answering (QA) applications, most notably exemplified by ChatGPT [1], which has quickly become integrated into the daily lives of internet-savvy users. With continuous updates, these models have expanded their capabilities—from writing code and interpreting documents to generating images in the style of various artists—thus broadening their range of real-world applications. However, biomedical knowledge, like other domain-specific expertise, has not been as seamlessly incorporated into this wave of general AI adoption. Due to its highly specialized nature, biomedical information is often difficult for the general public to verify. When an LLM provides incorrect or misleading answers in this context, users may make poor decisions with potentially harmful consequences. This concern is exacerbated by the growing number of users who rely solely on ChatGPT for information, forgoing deeper searches or expert consultation.

To advance biomedical QA for general users, it is essential to ensure that responses are not only accurate but also grounded in reliable sources. With the continuous accumulation of biomedical literature, there exists a rich and ever-growing resource for information retrieval and QA. Notably, medical knowledge is subject to change, as newer studies may refute previous findings—highlighting the importance of using up-to-date references. The annually updated PubMed baseline[1] provides a stable and comprehensive source of biomedical publications. However, such scientific literature is often inaccessible to non-experts due to its technical language. Moreover, raw documents without domain-specific annotations are difficult to leverage in QA systems. This discrepancy explains why

CLEF 2025 Working Notes, 9 -- 12 September 2025, Madrid, Spain

[*] Corresponding author.

joyhan@cc.ncu.edu.tw (J. Han); charlie963258@gmail.com (B. Chih); cgh23991@cgh.org.tw (H. Hung); thtsai@g.ncu.edu.tw (R. T. Tsai)

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

large-scale, high-quality datasets for general-domain QA are easier to construct, whereas datasets in specialized domains like biomedicine are produced at a much slower pace—particularly in the case of clinical records, which require anonymization and expert curation. Since 2012, the BioASQ Challenge [2-5] has addressed this gap by providing annotated datasets for biomedical information retrieval and QA. Each year's test set builds upon previous ones, forming a cumulative benchmark that drives progress in the field. Importantly, BioASQ incorporates both automated and manual evaluation methods. In some cases, manual assessment of answers has led to ranking shifts among participating systems, indicating that current automatic metrics do not fully capture human-level comprehension or answer quality. As biomedical QA systems become more robust, they hold the potential to assist medical professionals in making evidence-based decisions and uncovering latent knowledge from the latest literature.

To enable LLMs to provide authoritative answers, it is essential to ground their outputs in verifiable and citable sources. This is precisely the goal of retrieval-augmented generation (RAG) [6], a technique that enhances the accuracy and reliability of generative AI by incorporating information retrieved from relevant, domain-specific sources. RAG has seen widespread adoption, reflecting both the current capabilities and future direction of generative AI systems. In this work, we explore its potential in the biomedical domain by participating in the BioASQ 13b Challenge, specifically focusing on Phase A and A+—tasks that involve document retrieval and answer generation. Our system adopts a RAG-based architecture comprising three main components: a retriever, a reranker, and an LLM for natural language generation (NLG). For the retriever, we employ the classical information retrieval method BM25 to index the PubMed 2024 baseline dataset. From this index, we retrieve the top 100 documents relevant to each question. These candidates are then re-ranked using the BAAI/bge-reranker-v2-m3 model, which is optimized for QA tasks, to identify the most relevant documents or snippets. For answer generation, we utilize two models: the open-source meta-llama/Llama-3.1-8B-Instruct and the commercial GPT-4o. We experiment with two different types of input to the generation stage: full documents and extracted snippets. The document-based input allows for end-to-end generation directly from retrieved texts, while the snippet-based input is integrated with Chih et al.'s [7] previously developed Phase B system, which was designed to answer questions based on manually annotated snippets. This dual-input approach allows us to assess the impact of input granularity on generation efficiency and answer quality.

2. Related Work

From the perspective of the core architecture adopted in this study, RAG is a technical framework that integrates LLMs with external knowledge retrieval to enhance the accuracy of QA and content generation [6]. Notably, multiple teams employed RAG-based systems in last year's BioASQ 12b challenge [7-12], demonstrating the framework's effectiveness in the biomedical QA domain. RAG consists of two main components: a retriever and a generator. In simple terms, before generating an answer, the system must first retrieve relevant information from external sources through a three-step process: indexing, retrieval, and generation. During the indexing phase, external data are processed—typically through tokenization, vectorization, or other techniques—and stored in a searchable database. In the retrieval phase, the user's question is compared against this database to identify the most relevant documents. These documents, along with the original question, are then fed into the LLM to generate a final answer. RAG effectively links external resources to generative AI models, functioning like in-text citations during a conversation. This approach helps reduce hallucinations—plausible-sounding but incorrect outputs—by grounding responses in real, retrievable sources. Moreover, because RAG-based systems retrieve information from an external and continually updatable knowledge base, their knowledge is not limited to a static training set. This enables them to incorporate the latest information over time, ensuring that answer quality does not degrade due to outdated knowledge. In the biomedical domain, this dynamic and reference-based approach offers the potential for LLMs to act as reliable assistants to professionals, supporting decision-making with up-to-date and verifiable evidence.

As the FlashRAG Toolkit [13] introduces a more modular pipeline that allows for the flexible integration of components tailored to specific needs, we opted for a retriever-reranker setup, using BM25 [14] for sparse retrieval and pairing it with a dedicated reranker module. BM25, a classic term-frequency–based sparse retrieval method, remains a widely used [9-12, 15-17] and computationally efficient approach, particularly suitable for large-scale corpora with limited computational resources. While neural dense retrieval models such as BERT-based encoders are capable of capturing semantic similarity, they often fall short in precise lexical matching, an area where BM25 excels. To further improve retrieval precision, we incorporate a reranker to re-evaluate the top-ranked documents retrieved by BM25. This component helps determine whether a document contains a snippet that directly answers the user's question, thereby improving answer relevance. Based on comparative results from the previous year's Batch 1 test set, we selected BAAI/bge-reranker-v2-m3 as our reranker. This model—part of the M3-Embedding framework [18]—unifies several retrieval paradigms, including dense retrieval, lexical (sparse) retrieval, and multi-vector retrieval. Notably, it employs a novel self-knowledge distillation strategy, where relevance signals from multiple retrieval modes are integrated as teacher supervision to improve training robustness. The reranker demonstrates strong performance in both monolingual and cross-lingual retrieval tasks. Moreover, its lightweight design and fast inference speed make it well-suited for practical deployment, and it performed smoothly in our experiments.

After the retrieval stage, we transitioned to the answer generation phase. Given computational constraints, we primarily adopted the open-source meta-llama/Llama-3.1-8B-Instruct[2] model to balance generation quality with low-latency inference. As an upgraded 8B parameter model, Llama-3.1-8B [19] supports multilingual capabilities, offers a significantly extended context window of up to 128K tokens, and features enhanced tool usage and stronger reasoning abilities overall. Benchmark results have shown that it outperforms GPT-3.5 Turbo in multiple tasks. While it does not yet surpass the latest frontier models, it provides sufficient performance for constrained generation scenarios. During the release period of the BioASQ test sets, we also incorporated GPT-4o[3] into selected configurations beginning with Batch 2, enabling a comparison between an open-source and a proprietary model. GPT-4o matches GPT-4 Turbo in English text and code performance, while also offering improved speed and multimodal capabilities. As previous teams have already explored the GPT family in BioASQ tasks [7-11, 17, 20], we were particularly interested in comparing how biomedical QA performs under these two generation backbones.

3. Method

In this section, we provide a step-by-step overview of the corpus and task dataset, system architecture, components, LLM pipelines, and the configurations of the submitted systems.

3.1. Corpus and Dataset

For the BioASQ 13b challenge, the Phase A and A+ training datasets consist of 5,389 QA pairs. These included 1,459 yes/no questions, 1,600 factoid questions, 1,047 list-type questions, and 1,283 summary questions. Each question was accompanied by a list of relevant documents, relevant snippets (extracted from those documents), an exact answer (except for summary questions), and an ideal answer.

All associated documents and snippets were derived from the PubMed baseline corpus. The version of the PubMed baseline used in this year's competition was released at the end of 2024, containing a total of 38,201,553 documents. We indexed this corpus with a BM25-based retriever to enable document retrieval. Out of the full corpus, 38,178,296 documents were successfully indexed, while 23,257 entries were empty and excluded from retrieval.

[2] https://ai.meta.com/blog/meta-llama-3-1/

[3] https://openai.com/index/hello-gpt-4o/

3.2. System Overview

The overall RAG workflow of our system is illustrated in **Figure 1**. It consists of three main components: consists of three main components: a retriever, a reranker, and an LLM-based answer generation. The user question is fed into all three components ensuring that each stage in the pipeline has direct access to the original question for context-aware processing. Depending on the output of the reranker, the system branches into two distinct pipelines based on the types of retrieved content: either document or snippet. In Pipeline A, the top 10 documents are directly fed into the LLM for end-to-end answer generation. In Pipeline B, the top 10 snippets are selected instead, and then processed through the answer generation pipeline previously developed by Chih et al. [7], which was originally designed for snippet-based QA tasks (BioASQ b Phase B).

In the following sections, we provide a more detailed explanation of each component and the two pipeline variants.

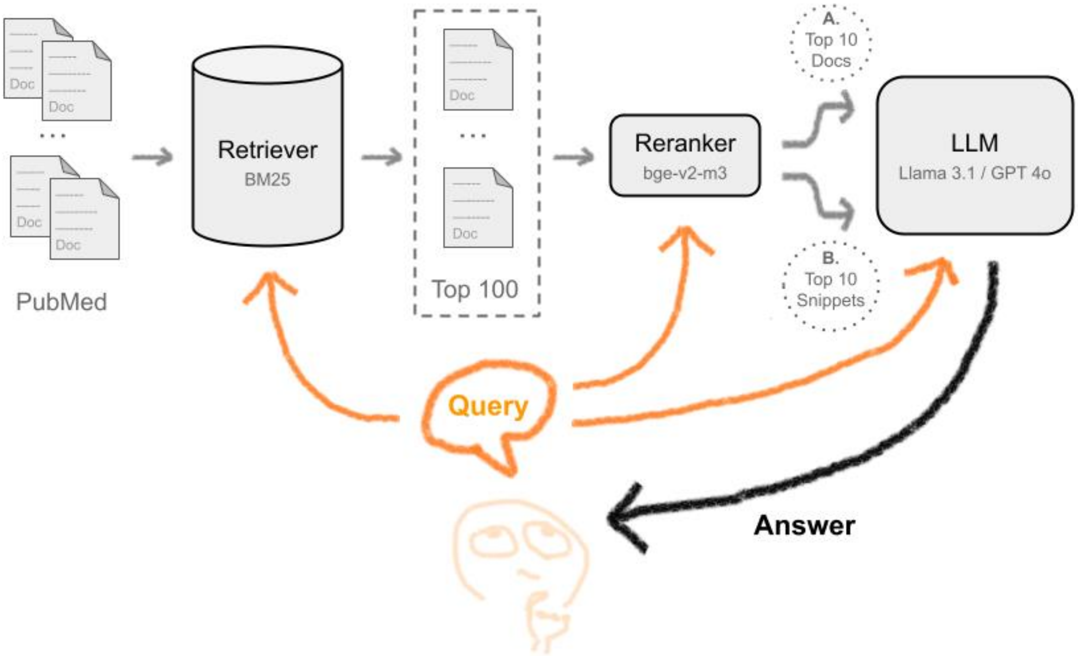

Figure 1: The RAG workflow of our system.

3.3. Retriever

For the retriever component, we adopted BM25 as a reliable lexical-based retrieval baseline, using the 2024 PubMed baseline as our document source. We implemented our RAG system using the Python-based FlashRAG toolkit[4]. To streamline processing, each PubMed article was simplified into two fields: the article ID and its content (title and abstract). Our BM25-based retriever was implemented using the Pyserini [21] Python toolkit within the FlashRAG framework, and we used the FlashRAG's default parameters. The retriever first identifies the top 100 documents most relevant to the input question, which are then passed to the reranker for further processing.

3.4. Reranker

To select our reranker component, we evaluated several open-source rerankers using the BioASQ 12b Phase A Batch 1 test set, as summarized in **Table 1** (see next page). The baseline retrieval was performed using BM25 over the PubMed 2023 baseline, since the BioASQ 12b does not include articles from PubMed 2024. We applied each reranker to the same top 100 documents retrieved by BM25. Among the BGE series models, BAAI/bge-reranker-v2-m3 achieved the best performance and

[4] https://github.com/RUC-NLPIR/FlashRAG

was thus selected as our primary single reranker. To further explore the effectiveness of reranker ensembles, we combined the top three rerankers from different sources—BAAI/bge-reranker-v2-m3 [18], mixedbread-ai/mxbai-rerank-large-v1 [22], and Alibaba-NLP/gte-reranker-modernbert-base [23]. Their scores were first normalized to a 0–1 range before being aggregated and re-ranked.

Table 1
The evaluation results of different rerankers on the BioASQ 12b Phase A Batch 1 Test Set (ranked by MAP).

Reranker	Mean precision	Recall	F-Measure	**MAP**	GMAP
BAAI/bge-reranker-v2-m3 [18]	0.4212	0.303	0.2913	0.3857	0.1736
mixedbread-ai/mxbai-rerank-large-v1 [22]	0.4235	0.3042	0.2944	0.3824	0.1691
mixedbread-ai/mxbai-rerank-base-v1 [22]	0.3976	0.2835	0.2724	0.3762	0.1432
Alibaba-NLP/gte-reranker-modernbert-base [23]	0.3988	0.2868	0.2751	0.3641	0.1571
Alibaba-NLP/gte-multilingual-reranker-base [23]	0.4012	0.2923	0.2807	0.3596	0.1302
BAAI/bge-reranker-large [24]	0.4012	0.2923	0.2807	0.3596	0.1302

From the top 100 retrieved documents, the reranker then selects either the top 10 documents or the top 10 snippets (obtained through sentence segmentation of those documents) depending on the specific requirements of the BioASQ 13b Phase A task. We submitted three different IR configurations for this phase: one system using only the retriever (IR5), one incorporating the single reranker (IR1), and another using the ensemble of three rerankers (IR4). A summary of the submitted systems for Phase A is shown in **Table 2**. The selected top 10 documents or top 10 snippets were then used as input to the LLM component, forming two separate pipeline branches, which will be detailed in the following sections.

Table 2
Overview of the submitted systems for BioASQ 13b Phase A. For each question, we first retrieved the top 100 relevant documents. In the Pipeline A, the top 10 documents were selected as the official Phase A document retrieval results. In the Pipeline B, we extracted all sentences from the top 100 documents and selected the top 10 sentences to serve as the snippet results for the Phase A.

Output		System		
		IR1	IR4	IR5
Documents	Step 1. Top 100	BM25	BM25	BM25
	Step 2. Top 10 (for Pipeline A)	Reranker	3 reranker	BM25
Snippets	Top 10 (after Docs Step 1.) (for Pipeline B)	Reranker	3 reranker	-

3.5. LLM

Due to limited computational resources, our system was primarily developed using the open-source meta-llama/Llama-3.1-8B-Instruct, with GPT-4o integrated during the testing phase to enhance the answer generation performance. All experiments were run on a machine with one NVIDIA RTX 3090 and one GTX 1080 GPU. Depending on the type of input selected during the IR stage, we employed different generation pipelines: Pipeline A used the top 10 retrieved documents as input, while Pipeline B used the top 10 extracted snippets.

3.5.1. Pipeline A

To ensure consistency in our experiment, we configured the LLM with a temperature of 0, aiming to produce deterministic outputs. Additionally, we ensured that the output length was sufficient to avoid incomplete responses.

When using documents as input, we designed prompts for generating both exact and ideal answers for Phase A+, as illustrated in **Table 3**. To avoid potential information loss due to long prompt length when concatenating the top 10 documents, each document was fed into the LLM separately. As a result, each user question has 10 exact answers and 10 ideal answers.

For exact answers, we applied a simple aggregation strategy: we selected the most frequently generated answer among the 10 outputs. In the case of a tie, the answer from the earliest ranked document (preserving the original document order) was chosen as the final output.

For ideal answers, simple majority voting was not applicable due to possible variations in phrasing. Instead, we concatenated the 10 generated ideal answers and used the LLM again to select the response that best addressed the question.

Table 3
Prompt templates used in Pipeline A for generating exact and ideal answers. Each of the top 10 retrieved documents is provided to the LLM separately along with the user question and its type (e.g., yes/no, factoid).

Role	Content
system	Generate a **JSON** response with the following structure. Ensure that both "exact_answer" and "ideal_answer" fields are always included in the output: 1. "exact_answer": Provide a response based on the question type. If the provided document does not contain enough information to answer, return an empty string (""): - **Yes/No Questions**: Answer with either "yes" or "no". - **Factoid Questions**: Provide a specific entity name (e.g., disease, drug, gene), a number, or a similar short expression. - **List Questions & Multiple Choice Questions**: Provide a list of entity names (e.g., gene names), numbers, or similar short expressions. If the model generates a comma-separated string instead of a list, convert it into a list format. - **Summary Questions**: Return an empty string ("") since these questions do not require an exact answer. 2. "ideal_answer": Generate a concise summary of the most relevant information. Follow these constraints: - **Word Limit**: The response **must not exceed 200 words** under any circumstances. - **Content Limitation**: Only extract and summarize information from the document; do **not** add any personal reasoning, assumptions, or explanations. - **No Guessing**: If the document does **not** provide enough relevant information, **return an empty string ("") instead of attempting to answer**. Strictly base the answer on the provided document:
system	[Document]
user	[Question Type] question: [Question] Answer:

3.5.2. Pipeline B

In this pipeline, the LLM input consists of the top 10 snippets, following a process similar to that used in BioASQ Task b Phase B. However, while Phase B provides expert-annotated snippets, Phase A+ relies on pseudo-snippets derived from retrieved documents. To explore this setting further, we extended our system in the later stage of the competition by incorporating the framework developed by Chih et al. [7], which also leverages RAG techniques. This extended system was paired with GPT-4o and used for comparison with our primary design in Pipeline A.

Table 4 (see next page) summarizes the configurations of our submitted systems for BioASQ 13b Phase A+, categorized by pipeline type and the LLM used.

Table 4
Overview of the submitted systems for BioASQ 13b Phase A+. During the competition period, any newly added IR configurations were paired with GPT-4o as the default choice for the NLG component.

Pipeline	IR Modul	NLG Modul	
		Llama-3.1-8B-Instruct	**GPT 4o**
A: Top 10 Documents	BM25	IR5	-
	BM25 + Reranker	IR1	IR2
B: Top 10 Snippets	Reranker (from Sentences of BM25 Top 100 Docs)	-	IR3
	3 Reranker (from Sentences of BM25 Top 100 Docs)	-	IR4

4. Results

Since our system was developed concurrently with the competition timeline, we faced server issues during the early stages. As a result, complete system submissions began from Batch 3. Starting with Batch 3, we submitted three systems for BioASQ 13b Phase A (IR1, IR4, IR5) and five systems for Phase A+ (IR1-IR5).

This section shows our preliminary results of BioASQ Task 13b. The final and official results will be released in September, following the manual evaluation of all system responses by BioASQ experts and the enrichment of the ground truth with potentially additional correct answers. As such, rankings and final scores are not reported in this paper and the current results are for reference only.

4.1. Phase A Results

The results for BioASQ 13b Phase A are presented in **Table 5**. Overall, our system performed above the median across both document and snippet retrieval tasks. Even for IR5, our BM25-only baseline, the scores were generally around the median which suggests that BM25 remains a widely adopted and dependable approach for document retrieval among participating teams. Further improvements were observed with IR1 and IR4, both of which incorporated a reranker after initial retrieval. This shows the reranker's benefit in refining the relevance ranking between the user question and candidate documents. However, IR4, which employed an ensemble of three different reranker variants, did not outperform IR1, indicating that such ensembles may bring noise issues rather than improve ranking quality. A single, strong reranker (IR1) achieved better results. As for snippet retrieval, while our systems still performed above the median, the gap between our best runs and the top-performing systems was more obvious. This suggests that our current strategy which ranks individual sentences from the retrieved documents solely with a reranker remains insufficient. More sophisticated techniques may be necessary to improve snippet-level retrieval performance further.

Table 5
The results of the submissions for BioASQ 13b Phase A. Only the primary evaluation metrics are reported here; for the complete set of scores, please refer to the official leaderboard. The dense rank for each MAP score is provided directly below the corresponding value in the table.

System	MAP							
	Batch 1		Batch 2		Batch 3		Batch 4	
	Documents	Snippets	Documents	Snippets	Documents	Snippets	Documents	Snippets
Best	0.4246	0.4535	0.4425	0.5522	0.3236	0.4322	0.1801	0.1634
IR1	0.3394 *(13/46)*	-	0.3548 *(12/38)*	0.2638 *(13/32)*	0.2665 *(11/38)*	0.2167 *(7/29)*	0.1586 *(8/65)*	0.0951 *(10/46)*
IR4	-	-	-	-	0.2344 *(13/38)*	0.2164 *(8/29)*	0.1333 *(16/65)*	0.0922 *(11/46)*
IR5	-	-	0.3225 *(15/38)*	-	0.2099 *(17/38)*	-	0.0751 *(27/65)*	-
Median	0.2527	0.1085	0.2986	0.2012	0.1834	0.0968	0.0626	0.0239

4.2. Phase A+ Results

The results for BioASQ 13b Phase A+ are shown in **Table 6**. Our system consistently achieved above-median performance on ideal answers, indicating that LLMs are effective in generating long answers compared to the more volatile results in exact answers. Among systems using the Llama-3.1-8B-Instruct model, the IR5 baseline which relies solely on BM25 performed worse across all metrics compared to IR1, which incorporated a reranker after BM25 retrieval. This highlights the importance of reranking in improving overall performance. Comparing IR1 and IR2, both of which used the top 10 documents (Pipeline A), we observed further improvement in ideal answers when using GPT-4o. However, the performance was closed for exact answers between GPT-4o and Llama-3.1-8B-Instruct. IR3, which also used GPT-4o but with the top 10 snippets (Pipeline B), showed additional gains in Ideal answers. This suggests that integrating the snippet-based approach from Chih et al. remains beneficial, although the impact varies by question type. In contrast, IR4 (despite also using GPT-4o) utilized an ensemble of three different reranker models to select the top 10 snippets. The results showed no consistent advantage over the single reranker setup in IR3, implying that reranker ensembles may introduce noise rather than improve reliability.

Table 6
The results of the submissions for BioASQ 13b Phase A+. The table below presents the evaluation results for both exact and ideal answers. Only the primary evaluation metrics are reported here; for the complete set of scores, please refer to the official leaderboard. Each evaluation score is accompanied by its dense rank, shown directly below the value in the table. **Bolded values** indicate that our system achieved the highest score for that specific question type. *Italicized values* denote the best-performing system submitted by our team for batches 3 and 4.

Batch	Answer Type	Q Type	Evaluation	System						
				Best	**IR1**	**IR2**	**IR3**	**IR4**	**IR5**	Median
1	Exact	Yes/No	Macro F1	1.0000	0.8132 *(6/12)*	-	-	-	-	0.9244
		Factoid	MRR	0.4551	0.3846 *(4/24)*	-	-	-	-	0.2885
		List	F-Measure	0.2567	0.1306 *(32/47)*	-	-	-	-	0.14215
	Ideal	all	R-SU4 (F1)	0.2133	0.1461 *(10/52)*	-	-	-	-	0.11175
2	Exact	Yes/No	Macro F1	1.0000	-	0.7018 *(12/17)*	-	-	-	0.8132
		Factoid	MRR	0.5926	-	0.4444 *(8/27)*	-	-	-	0.3519
		List	F-Measure	0.3880	-	0.1974 *(31/40)*	-	-	-	0.233
	Ideal	all	R-SU4 (F1)	0.2177	-	0.2040 *(7/46)*	-	-	-	0.1263
3	Exact	Yes/No	Macro F1	0.9394	0.6179 *(15/22)*	**0.9394** *(1/22)*	0.6944 *(11/22)*	0.5417 *(19/22)*	0.6071 *(16/22)*	0.7412
		Factoid	MRR	0.3750	0.1500 *(18/23)*	0.1000 *(21/23)*	*0.3500* *(2/23)*	0.2750 *(9/23)*	0.1500 *(18/23)*	0.2
		List	F-Measure	0.4541	0.3482 *(18/48)*	0.3220 *(26/48)*	*0.4313* *(4/48)*	0.3632 *(12/48)*	0.2686 *(32/48)*	0.2902
	Ideal	all	R-SU4 (F1)	0.2085	0.1553 *(14/55)*	0.1828 *(11/55)*	*0.2058* *(2/55)*	0.2013 *(5/55)*	0.1424 *(21/55)*	0.11875
4	Exact	Yes/No	Macro F1	0.9097	0.8194 *(7/21)*	0.7815 *(11/21)*	*0.8595* *(4/21)*	0.8194 *(7/21)*	0.6601 *(19/21)*	0.8194
		Factoid	MRR	0.5606	0.4091 *(10/25)*	*0.5000* *(5/25)*	0.4318 *(8/25)*	0.3636 *(13/25)*	*0.5000* *(5/25)*	0.3788
		List	F-Measure	0.3014	0.2544 *(15/57)*	0.2845 *(7/57)*	*0.2918* *(5/57)*	0.2492 *(19/57)*	0.2196 *(36/57)*	0.227
	Ideal	all	R-SU4 (F1)	0.1726	0.1332 *(17/65)*	0.1467 *(10/65)*	0.1553 *(6/65)*	*0.1665* *(2/65)*	0.1202 *(24/65)*	0.0986

5. Conclusion

Based on the preliminary results, BM25-based retriever remains a reliable performance for information retrieval in Phase A. When combined with a reranker, performance improves further—

especially in document retrieval—though there is still room for enhancement. However, for snippet retrieval, the current setup remains underdeveloped and requires significant improvement.

In the NLG stage (Phase A+), both meta-llama/Llama-3.1-8B-Instruct and GPT-4o demonstrate strong performance in generating ideal answers. GPT-4o tends to outperform Llama-3.1-8B-Instruct on average when using the same document inputs (Pipeline A). Moreover, when GPT-4o is provided with snippet-based inputs (Pipeline B) and a more structured generation pipeline, scores improve even further. That said, for exact answers, the performance between the two LLMs varies by question type and appears highly dependent on the annotations in each batch.

This competition provided valuable insights into the inherent challenges of the BioASQ task. From a data perspective, the annual one-time update of the PubMed baseline—unlike the continuously updated files that include new, revised, and deleted citations—poses a significant challenge. Earlier BioASQ questions may have been annotated based on documents that have since been modified or removed, making it more difficult for models using the most recent PubMed baseline to answer older questions accurately. Combined with the diverse nature of the questions and the subjective variability introduced by different annotators, maintaining stable model performance in BioASQ is particularly difficult. We experienced this firsthand when attempts to fine-tune a reranker using BioASQ's training data failed to converge during the competition.

Acknowledgements

This research was supported by the Ministry of Science and Technology, Taiwan (112-2221-E 008-062-MY3).

Declaration on Generative AI

During the preparation of this work, the author used ChatGPT in order to: Text Translation. After using this service, the author reviewed and edited the content as needed and takes full responsibility for the publication's content.

References

[1] J. Achiam *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774,* 2023.

[2] G. Tsatsaronis *et al.*, "An overview of the BIOASQ large-scale biomedical semantic indexing and question answering competition," *BMC bioinformatics,* vol. 16, pp. 1-28, 2015.

[3] A. Krithara, A. Nentidis, K. Bougiatiotis, and G. Paliouras, "BioASQ-QA: A manually curated corpus for Biomedical Question Answering," *Scientific Data,* vol. 10, no. 1, p. 170, 2023.

[4] A. Krithara, J. G. Mork, A. Nentidis, and G. Paliouras, "The road from manual to automatic semantic indexing of biomedical literature: a 10 years journey," *Frontiers in Research Metrics and Analytics,* vol. 8, p. 1250930, 2023.

[5] A. Nentidis *et al.*, "Overview of BioASQ 2025: The thirteenth BioASQ challenge on large-scale biomedical semantic indexing and question answering," in *Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Sixteenth International Conference of the CLEF Association (CLEF 2025),* J. Carrillo-de-Albornoz *et al.*, Eds., 2025.

[6] P. Lewis *et al.*, "Retrieval-augmented generation for knowledge-intensive nlp tasks," *Advances in neural information processing systems,* vol. 33, pp. 9459-9474, 2020.

[7] B.-C. Chih, J.-C. Han, and 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.

[8] S. Ateia and 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.

[9] Y. Gao, L. Zong, and Y. Li, "Enhancing biomedical question answering with parameter-efficient fine-tuning and hierarchical retrieval augmented generation," *CLEF Working Notes,* 2024.

[10] B.-W. Huang, "Generative large language models augmented hybrid retrieval system for biomedical question answering," *CLEF Working Notes,* 2024.

[11] J. H. Merker, A. Bondarenko, M. Hagen, and A. Viehweger, "MiBi at BioASQ 2024: retrieval-augmented generation for answering biomedical questions," in *Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2024), Grenoble, France*, 2024, vol. 3740, pp. 176-187.

[12] D. Panou, A. Dimopoulos, and M. Reczko, "Farming open LLMs for biomedical question answering," *CLEF Working Notes,* 2024.

[13] J. Jin *et al.*, "Flashrag: A modular toolkit for efficient retrieval-augmented generation research," *arXiv preprint arXiv:2405.13576,* 2024.

[14] S. Robertson and H. Zaragoza, "The probabilistic relevance framework: BM25 and beyond," *Foundations and Trends® in Information Retrieval,* vol. 3, no. 4, pp. 333-389, 2009.

[15] T. Almeida, R. A. Jonker, J. Reis, J. R. Almeida, and S. Matos, "BIT. UA at BioASQ 12: From Retrieval to Answer Generation," 2024.

[16] M. Lesavourey and G. Hubert, "Enhancing Biomedical Document Ranking with Domain Knowledge Incorporation in a Multi-Stage Retrieval Approach," in *12th BioASQ Workshop at CLEF 2024*, 2024, vol. 3740.

[17] O. Şerbetçi, X. D. Wang, and U. Leser, "HU-WBI at BioASQ12B Phase A: Exploring Rank Fusion of Dense Retrievers and Re-rankers," in *Proceedings of the Conference and Labs of the Evaluation Forum, Grenoble, France*, 2024, pp. 9-12.

[18] J. Chen, S. Xiao, P. Zhang, K. Luo, D. Lian, and Z. Liu, "M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation," in *Findings of the Association for Computational Linguistics ACL 2024*, 2024, pp. 2318-2335.

[19] A. Grattafiori *et al.*, "The llama 3 herd of models," *arXiv preprint arXiv:2407.21783,* 2024.

[20] W. Zhou and T. H. Ngo, "Using pretrained large language model with prompt engineering to answer biomedical questions," *arXiv preprint arXiv:2407.06779,* 2024.

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

[22] A. Shakir, D. Koenig, J. Lipp, and S. Lee, "Boost your search with the crispy mixedbread rerank models," ed, 2024.

[23] X. Zhang *et al.*, "mgte: Generalized long-context text representation and reranking models for multilingual text retrieval," *arXiv preprint arXiv:2407.19669,* 2024.

[24] S. Xiao, Z. Liu, P. Zhang, N. Muennighoff, D. Lian, and J.-Y. Nie, "C-pack: Packed resources for general chinese embeddings," in *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*, 2024, pp. 641-649.