NCU-IISR: Biomedical Question Answering via Gemini and GPT APIs in the BioASQ 13b Phase B Challenge

Bing-Chen Chih[1,*,†], Jen-Chieh Han[1], Hsi-Chuan Hung[2] and Richard Tzong-Han Tsai[1,2,3,*,†]

[1]Department of Computer Science and Information Engineering, National Central University, Taiwan
[2]Department of Medical Research, Cathay General Hospital, Taipei, Taiwan
[3]Research Center for Humanities and Social Sciences, Academia Sinica, Taiwan

Abstract

In this paper, we present our system and submissions for the BioASQ 13b Phase B task, continuing our efforts to improve biomedical question answering (QA) using large language models (LLMs). Building on prior work, we explored the integration of Google's Gemini API alongside OpenAI's Chat Completions API to compare and leverage the strengths of both models in the biomedical domain. Our system retains the use of Retrieval-Augmented Generation (RAG) techniques by employing file-based contextual search to retrieve relevant background documents, which are then incorporated into model prompts. We applied refined prompt engineering strategies tailored for factoid, list, and yes/no questions. Through comprehensive experiments with both LLM APIs, we identified optimal prompting patterns and response consolidation methods. Our final submission utilized a multi-model pipeline and achieved competitive results across multiple evaluation metrics, demonstrating the effectiveness of multi-model orchestration and document-grounded generation in biomedical QA.

Keywords

Biomedical Question Answer, Large Language Models (LLMs), Generative Pre-trained Transformer, Gemini, Retrieval Augmented Generation, CEUR-WS

1. Introduction

The BioASQ shared task [1] has been a leading benchmark for advancing biomedical semantic indexing and question answering through its annual shared tasks since 2013. In its 13th iteration, Task 13b Phase B challenges participants to generate both exact and ideal answers to biomedical questions using provided textual snippets. The 2025 dataset[2] consists of 5,389 questions, comprising prior annotations with gold-standard answers and 340 newly curated test questions. These are organized into four batches of 85 questions each, crafted by domain experts. The task covers four question types: yes/no, factoid, list, and summary. While all types require an ideal answer, only yes/no, factoid, and list questions require exact answers. Participants are allowed up to five submissions per batch, facilitating iterative refinement of their QA systems.

Each instance in the BioASQ dataset includes a question, one or more relevant snippets, and corresponding gold answers categorized into "ideal" and "exact" forms. Table 1 provides representative examples for each question type. In previous work [3], we achieved competitive performance by leveraging GPT-4's language understanding capabilities in combination with tailored prompt engineering strategies and Retrieval-Augmented Generation (RAG) techniques [4].

In 2025, we extend our approach by developing multiple independent systems based on a single large language model. Specifically, we experiment with gpt-4o[5] and o3-mini[6] from OpenAI, as well as gemini-2.0-flash[7] from Google. Each system is designed to operate separately without combining outputs from different models. All systems continued to utilize an RAG framework, employing file-based document search to retrieve relevant biomedical content, which is incorporated into prompt construction to enhance factual grounding. Prompt engineering techniques are further refined for each

CLEF 2025 Working Notes, 9 – 12 September 2025, Madrid, Spain
*Corresponding author.
†These authors contributed equally.
bingchen0714@gmail.com (B. Chih); joyhan@cc.ncu.edu.tw (J. Han); cgh23991@cgh.org.tw (H. Hung); thtsai@g.ncu.edu.tw (R. T. Tsai)

question type to match the strengths and limitations of the underlying model. This setup allows for a controlled comparison of LLMs under consistent conditions in the BioASQ 13b Phase B task.

Table 1
Examples across four categories in the BioASQ dataset

Yes/No	
Question	Can modulation of KCNQ1 splicing prevent arrhythmias?
Exact Answer	yes
Ideal Answer	Amiloride reduces arrhythmogenicity through the modulation of KCNQ1 splicing. Therefore, the modulation of KCNQ1 splicing may help prevent arrhythmias.
List	
Question	Which drugs are included in the AZD7442?
Exact Answer	[tixagevimab, cilgavimab]
Ideal Answer	AZD7442 is a combination of two long-acting monoclonal antibodies tixagevimab and cilgavimab. It has been authorized for the prevention and treatment of coronavirus disease 2019 (COVID-19).
Factoid	
Question	Olokizumab is tested for which disease?
Exact Answer	[rheumatoid arthritis]
Ideal Answer	Olokizumab, a monoclonal antibody against interleukin 6, improves outcomes of rheumatoid arthritis.
Summary	
Question	What is the definition of dermatillomania?
Ideal Answer	Dermatillomania is a condition that leads to repetitive picking of their skin ending up in skin and soft tissue damage. It is a chronic, recurrent, and treatment resistant neuropsychiatric disorder with an underestimated prevalence that has a concerning negative impact on an individual's health and quality of life.

2. Related Work

The biomedical domain presents significant challenges for information access due to its vast amount of domain-specific knowledge and complex terminology. Traditional information acquisition methods, such as manually reading large volumes of academic literature, are time-consuming and demand high professional expertise. This process often proves inefficient, particularly when medical professionals and the general public require rapid access to accurate biomedical information.

To address this inefficiency, Question Answering (QA) systems based on Natural Language Processing (NLP) have gained increasing attention. By leveraging large-scale language models, these systems can effectively interpret questions, retrieve relevant biomedical information, and generate accurate responses. This paradigm significantly improves the accessibility and usability of biomedical knowledge, bridging the gap between complex textual data and practical applications. With ongoing advances in deep learning, the performance of LLM-based QA systems continues to improve, enabling more effective support for biomedical research and clinical decision-making.

Prompt Engineering has become a pivotal strategy in optimizing the performance of large language models such as GPT, LLaMA, and their successors. It involves the deliberate design of input prompts to

elicit accurate and contextually appropriate outputs from pretrained models. Research has shown that carefully crafted prompts can significantly enhance model performance, particularly in few-shot[8] or zero-shot settings. For example, Brown et al.[8] demonstrated that prompt design substantially improved LLM accuracy in various NLP tasks with limited examples. This approach is now widely adopted across domains, including biomedical QA, summarization, and machine translation, where precise model behavior is critical.

Retrieval Augmented Generation integrates retrieval mechanisms with generative models to improve the factual accuracy and relevance of generated content. First introduced by Lewis et al.[4], RAG frameworks retrieve external documents relevant to a query and incorporate them into the generative process, guiding the model toward more informed and grounded outputs. This method has proven especially effective in open-domain and biomedical question answering, where domain-specific knowledge is crucial. Recent applications of RAG continue to demonstrate its advantages in enhancing response quality, especially when coupled with high-quality document retrieval systems and robust prompt integration techniques.

3. Methodology

We adopt an RAG framework to enhance answer quality in biomedical question answering. Traditionally, RAG consists of two components: a retriever, which identifies relevant documents based on the input question, and a generator, which uses those documents to generate responses. In prior work, we implemented a local retrieval pipeline using Dense Passage Retrieval (DPR), encoding both queries and documents into dense vectors for similarity-based matching.

This year, we simplify the retrieval process by utilizing the file search functionality provided by the respective LLM platforms, which effectively abstracts the retriever component while maintaining comparable retrieval relevance. Retrieved snippets are automatically appended to the prompt context, supporting accurate and grounded generation. This shift allows us to focus more on model behavior and prompt optimization without the overhead of maintaining a separate retriever infrastructure.

Each system in our setup employs only a single LLM. We constructed independent systems using OpenAI's gpt-4o, o3-mini, and Google's gemini-2.0-flash. We standardized the prompt structure, retrieval strategy, and output formatting across all systems.

3.1. Dataset

We used the BioASQ Task 13b Phase B dataset[2], which consists of 5,389 training samples derived from previous BioASQ tasks and newly added questions. The dataset includes four question types: summary (1,283), factoid (1,600), list (1047), and yes/no (1,459). Each question is paired with multiple snippets sourced from biomedical documents.

In contrast to our previous work, where only the top five snippets were retained due to token limitations, this year we incorporated all available snippets per question. Advances in LLM context length and API throughput enabled this expansion, we found that including all snippets enhanced model performance and led to more comprehensive answer generation, without degrading latency or fluency.

3.2. Prompt

Prompt Construction. We designed our prompting strategy based on two system variants:

- **Systems with file search:** For models that support file-based retrieval (e.g., `gpt-4o` using file search), the prompt contains only the **question itself**. Contextual information is automatically injected by the system based on the indexed document snippets.
- **Systems without file search:** In configurations that do not utilize file search (e.g., `o3-mini` and `gemini-2.0-flash`), we manually insert **all relevant snippets directly into the prompt**,

followed by the question. This allows the model to process contextual evidence inline without access to an external retriever.

Answer Generation. Each system is instructed to return both the *ideal* and *exact* answers in JSON format. In most configurations, both answers are generated in a single step. In an alternative two-stage pipeline, we first prompt the model to generate the ideal answer, followed by generating the exact answer using the ideal response as intermediate context. This strategy is motivated by the observation that entities in the exact answer often co-occur in the ideal response. To improve the accuracy of exact answers, we use few-shot examples for each question type.

Adaptation in Ideal Answer. In ideal answer, we observed a strong correlation between ideal answers and snippet phrasing. Accordingly, we modified prompts to encourage the model to reuse snippet segments in ideal answer verbatim. This adjustment led to improved fidelity and alignment with gold-standard references.

3.3. Strategy

Our approach employs two primary strategies for answer generation: (1) direct generation of both the ideal and exact answers in a single step, and (2) sequential generation in a two-stage format, where the ideal answer is first generated and subsequently used to guide the extraction of the exact answer. This design is conceptually aligned with the chain-of-thought paradigm [9], where intermediate reasoning enhances output precision. While two-stage prompting can improve structure and consistency—particularly for factoid and list questions—our experiments showed that single-stage prompting is often sufficient and more efficient in practice.

To enhance the factual grounding of generated answers, we integrated a RAG[4] approach by leveraging the file search functionality provided by the model platforms. This replaces the need for a custom embedding-based retriever. In systems with file search support (e.g., `gpt-4o`), relevant documents are uploaded and indexed beforehand; the model automatically incorporates pertinent content during inference. In contrast, for systems without file search capability (e.g., `o3-mini` and `gemini-2.0-flash`), we manually embed all relevant snippets into the prompt context.

Our dataset analysis observed that ideal answers frequently included verbatim segments from the supporting snippets. To exploit this pattern, we explicitly adjusted prompts to encourage snippet duplication in the generation process. This adjustment led to noticeable gains in factual accuracy and alignment with gold-standard answers. We also investigated the impact of decoding hyperparameters. Temperature settings were tuned across systems, with `temperature = 0` generally yielding more deterministic and concise responses. However, slight temperature increases in some cases improved fluency or prevented overly rigid outputs.

Overall, the combination of prompt refinement, strategic use of file search, and decoding control enabled our systems to produce contextually accurate and well-structured answers for the BioASQ 13b Phase B challenge.

Table 2 provides representative examples of our prompt templates for different stages and question types.

3.4. Systems

We employed different system configurations across batches to evaluate the performance of various language models and prompting strategies. Each system is based on a single model and follows the RAG-based prompting architecture described in previous sections. The detailed model assignments, prompting strategies, and use of file search for each batch are summarized in Table 3.

4. Result and Analysis

Our evaluation results are summarized in Table 4 (Exact Answer) and Table 5 (Ideal Answer). Our system did not demonstrate consistently strong performance in any single batch across the four evaluated

Table 2
The prompts using for generating ideal answer and exact answer separately

Tasks	Prompts
Ideal Answer	Reply to the answer clearly and easily in less than 3 sentences. You should read the chat history's content before answer the question. You can directly copy part of the above snippets as part of your answer. The question is: ""
Yes/No	Please answer me only "yes" or "no". You should read the reference's content before answer the question. The question is:
List	Please answer me and follow the following rules: 1. Give me a list of precise key entities to answer the question, as clear and concise as possible. 2. The list should contain entity names, jointly taken to constitute a single answer. 3. You should read the reference's content and chat history before answer the question. The Question is: ""
Factoid	Please answer me and follow the following rules: 1. Give me a list of precise key entities to answer the question, as clear and concise as possible. 2. The list should contain up to 5 entity names, ordered by decreasing confidence. 3. You should read the reference's content and chat history before answer the question. The Question is: ""

Table 3
Configuration of all submitted systems. Each system is defined by a unique model, a generation strategy (either direct generation of both answers or a two-stage approach), and whether Retrieval-Augmented Generation (RAG) via file search was applied.

Batch	System Name	Model	Generation Strategy	RAG
Batch-1	IISR-1	gemini-2.0-flash	Direct (Ideal + Exact)	None
	IISR-2	gemini-2.0-flash	Direct (Ideal + Exact)	File Search
	IISR-3	gemini-2.0-flash	Split (Ideal $\rightarrow$ Exact)	None
	IISR-4	gemini-2.0-flash	Split (Ideal $\rightarrow$ Exact))	File Search
	IISR-5	gpt-4o	Direct (Ideal + Exact)	File Search
Batch-2 to Batch-4	IISR-1	o3-mini	Direct (Ideal + Exact)	None
	IISR-2	gpt-4o	Direct (Ideal + Exact)	File Search
	IISR-3	o3-mini	Split (Ideal $\rightarrow$ Exact)	None
	IISR-4	gpt-4o	Split (Ideal $\rightarrow$ Exact))	File Search
	IISR-5	gpt-4o	Direct (Ideal + Exact)	None

batches. However, we observed relatively higher results in specific question types—namely *summary*, *yes/no*, and *list* questions—where several metrics showed stable and competitive performance.

Conversely, our systems consistently underperformed on *factoid* questions, regardless of the model or generation strategy applied. This suggests that further refinement is needed for entity-level answer extraction.

Among the various system configurations, those incorporating file-based retrieval (file search) demonstrated superior performance compared to those relying solely on in-prompt snippets. The added contextual grounding provided by file search enhances the model's ability to produce accurate and relevant responses, particularly for complex or abstract biomedical queries.

4.1. Key Findings and Cross-Model Behaviour

Across the four evaluation batches, gpt-4o with file search (IISR-2,-4,-5) achieved the most consistent strength on yes/no and list questions, achieving up to 0.94 accuracy for yes/no (Batch-2) and 0.64 F1 for list answers (Batch-3). These gains align with the intuition that explicit retrieval grounding mitigates hallucination and enables concise binary or enumerative responses. By contrast, o3-mini variants without retrieval (IISR-1,-3 in later batches) showed competitive yes/no accuracy however declined on list F1—reflecting limited token context and weaker factual grounding.

Factoid questions performance remained volatile for all models. We observed two recurring failure

Table 4
The Exact Answers test results on BioASQ. We define FIN scores as the average of Accuracy in Yes/No, MRR in Factoid, and F-Measure in List.

Batch	System	Yes/No		Factoid			List		
		Acc	maF1	SAcc	LAcc	MRR	Precision	Recall	F1
Batch-1	IISR-1	1.0000	1.0000	0.4231	0.4231	0.4231	0.5654	0.5538	0.5480
	IISR-2	1.0000	1.0000	0.3846	0.4615	0.4231	0.5694	0.6102	0.5798
	IISR-3	1.0000	1.0000	0.4231	0.5000	0.4615	0.5503	0.5328	0.5361
	IISR-4	**1.0000**	**1.0000**	0.4231	0.4615	0.4423	**0.5820**	**0.6224**	**0.5959**
	IISR-5	0.9412	0.9328	**0.4615**	**0.5000**	**0.4808**	0.4004	0.3528	0.3720
Batch-2	IISR-1	**0.9412**	**0.9377**	0.5185	0.5556	0.5370	0.4961	0.5209	0.4954
	IISR-2	0.9412	0.9377	0.4444	0.4815	0.4630	0.4575	0.4522	0.4441
	IISR-3	0.9412	0.9377	0.5185	0.5185	0.5185	0.5263	0.5626	0.5281
	IISR-4	0.9412	0.9377	0.4444	0.4444	0.4444	**0.5905**	**0.5904**	**0.5800**
	IISR-5	0.8824	0.8712	**0.5556**	**0.5926**	**0.5741**	0.4960	0.5127	0.4903
Batch-3	IISR-1	0.9545	0.9394	0.3500	0.4000	0.3750	0.6048	0.5896	0.5781
	IISR-2	0.9091	0.8854	0.3000	0.3500	0.3250	0.6433	**0.6429**	**0.6337**
	IISR-3	**0.9545**	**0.9394**	**0.4000**	**0.4500**	**0.4250**	**0.6465**	0.6037	0.6069
	IISR-4	0.8636	0.8182	0.2000	0.2500	0.2250	0.6300	0.5924	0.5936
	IISR-5	0.9545	0.9394	0.2500	0.3000	0.2750	0.6357	0.6429	0.6261
Batch-4	IISR-1	**0.9231**	**0.9023**	0.4545	0.5000	0.4773	0.6137	0.6048	0.6061
	IISR-2	0.8077	0.7815	0.4545	0.4545	0.4545	**0.6443**	**0.6197**	**0.6259**
	IISR-3	0.8846	0.8595	**0.5000**	**0.5000**	**0.5000**	0.5686	0.5443	0.5531
	IISR-4	0.8846	0.8595	0.4091	0.4545	0.4318	0.4680	0.4008	0.4226
	IISR-5	0.8077	0.7815	0.4545	0.5000	0.4773	0.5758	0.5673	0.5679

modes:

- Entity granularity mismatch (e.g., returning drug class instead of molecule name)
- Partial answer coverage when the LLM stopped after one entity despite multiple correct options.

File search acts as a high-recall retriever that appends all relevant sections without occupying prompt tokens. Nevertheless, retrieval noise occasionally introduces spurious entities, which adversely affect factoid precision, underscoring the need for effective post-filtering.

4.2. Metric Anomaly Clarification

Across multiple batches (notably Batches 2 and 4) on factoid questions, some systems produced identical aggregate scores of SAcc = LAcc = MRR = 0.5000. This behaviour stems from two design choices and the way BioASQ computes factoid metrics:

1. **Single-entity output.** These systems emit exactly one entity per factoid question. In cases where the gold standard lists several synonyms (e.g., "tamoxifen", "TAM") or multiple distinct answers, our system proposes only the top candidate.
2. **Metric definitions.** For each question, BioASQ evaluates:
    - **SAcc** – strict accuracy, 1 if any returned entity matches the gold list, 0 otherwise.
    - **LAcc** – lenient accuracy, identical to SAcc when only one entity is returned.
    - **MRR** – reciprocal rank, equal to 1/r where r is the rank of the first correct entity (thus 1 when the sole answer is correct, 0 otherwise). Because these systems return a single item, these three per-question values are numerically identical.

5. Conclusions

In this year's participation, we explored multiple language model configurations and prompting strategies under a unified RAG-based question answering framework. While overall performance across the

Table 5
The Ideal Answers test results on BioASQ.

Batch	System	R-2 (Rec)	R-2 (F1)	R-SU4 (Rec)	R-SU4 (F1)
Batch-1	IISR-1	**0.4338**	**0.3883**	**0.4229**	**0.3681**
	IISR-2	0.4110	0.3562	0.4009	0.3385
	IISR-3	0.4295	0.3295	0.4173	0.3094
	IISR-4	0.4062	0.3054	0.3969	0.2838
	IISR-5	0.4147	0.3491	0.4070	0.3308
Batch-2	IISR-1	0.2991	0.2503	0.2863	0.2290
	IISR-2	**0.4307**	0.3875	**0.4114**	0.3598
	IISR-3	0.3349	0.2880	0.3194	0.2628
	IISR-4	0.3907	**0.3993**	0.3726	**0.3737**
	IISR-5	0.3977	0.3707	0.3824	0.3479
Batch-3	IISR-1	0.2247	0.1915	0.2350	0.1907
	IISR-2	**0.3787**	0.3351	**0.3777**	0.3248
	IISR-3	0.2954	0.2495	0.2978	0.2378
	IISR-4	0.3429	**0.3520**	0.3383	**0.3439**
	IISR-5	0.3762	0.3447	0.3774	0.3339
Batch-4	IISR-1	0.2261	0.1917	0.2276	0.1873
	IISR-2	0.3745	**0.3604**	0.3725	**0.3515**
	IISR-3	0.2850	0.2276	0.2852	0.2208
	IISR-4	0.3227	0.3433	0.3144	0.3345
	IISR-5	**0.3855**	0.3503	**0.3818**	0.3384

four batches did not surpass our previous year's results, we identified several areas of relative strength and weakness.

Our systems performed better on *summary*, *yes/no*, and *list* questions, indicating that the models were more effective at generating abstracted or binary content. On the other hand, *factoid* questions proved challenging across all models and configurations, suggesting limitations in precise entity extraction under current prompting schemes.

The use of file search for retrieval consistently contributed to improved results. This mechanism allowed for more flexible and scalable integration of contextual knowledge, and its impact was evident in higher scores for supported models. By contrast, systems that embedded all snippets directly into the prompt faced token and relevance limitations that may have affected accuracy.

Our findings suggest that better handling of factoid-style questions—potentially through entity-aware prompts or post-processing could be a critical direction for future improvement. Additionally, more adaptive retrieval methods and refined prompt segmentation strategies may further enhance the precision and reliability of biomedical QA systems.

Declaration on Generative AI

During the preparation of this work, the author used ChatGPT (GPT-4o) in order to: assist in drafting, perform grammar and style correction, and polish the text. After using this tool/service, the author reviewed and edited the content as needed and takes full responsibility for the publication's content.

References

[1] A. Nentidis, et al., Bioasq at clef2025: The thirteenth edition of the large-scale biomedical semantic indexing and question answering challenge, in: Advances in Information Retrieval. ECIR 2025. Lecture Notes in Computer Science, volume 15576, Springer, Cham, 2025. URL: https://doi.org/10.1007/978-3-031-88720-8_61. doi:10.1007/978-3-031-88720-8_61.

[2] A. Krithara, A. Nentidis, K. Bougiatiotis, G. Paliouras, BioASQ-QA: A manually curated corpus for Biomedical Question Answering, Scientific Data 10 (2023) 170.
[3] R. T. H. T. Bing Chen Chih, Jen Chieh Han, Ncu-iisr: Enhancing biomedical question answering with gpt-4 and retrieval augmented generation in bioasq 12b phase b, CEUR Workshop Proceedings 3740 (2024) 99–105.
[4] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, D. Kiela, Retrieval-augmented generation for knowledge-intensive nlp tasks, in: Advances in Neural Information Processing Systems, volume 33, 2020, pp. 9459–9474. URL: https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.
[5] OpenAI, Gpt-4o system card, 2024. URL: https://arxiv.org/abs/2410.21276. arXiv:2410.21276.
[6] OpenAI, OpenAI o3 and o4-mini System Card, Technical Report, OpenAI, 2025. URL: https://cdn.openai.com/pdf/2221c875-02dc-4789-800b-e7758f3722c1/o3-and-o4-mini-system-card.pdf, accessed: 2025-05-22.
[7] S. B. J.-B. A. e. a. Gemini Team, Rohan Anil, Gemini: A family of highly capable multimodal models, 2025. URL: https://arxiv.org/abs/2312.11805. arXiv:2312.11805.
[8] T. B. 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, arXiv preprint arXiv:2005.14165 (2020).
[9] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, D. Zhou, Chain-of-thought prompting elicits reasoning in large language models, 2023. arXiv:2201.11903.