NCU-IISR: Prompt Engineering on GPT-4 to Stove Biological Problems in BioASQ 11b Phase B

Chun-Yu Hsueh[1], Yu Zhang[1], Yu-Wei Lu[1], Jen-Chieh Han[1], Wilailack Meesawad[1] and Richard Tzong-Han Tsai[1,2,*]

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

Abstract

In this paper, we present our system applied in BioASQ 11b phase b. We showcase prompt engineering strategies and outline our experimental steps. Building upon the success of ChatGPT/GPT-4 in answer generation and the field of biology, we developed a system that utilizes GPT-4 to answer biomedical questions. The system leverages OpenAI's ChatCompletions API and combines Prompt Engineering methods to explore various prompts. In addition, we also attempted to incorporate GPT-4 into our system from last year, which combines a BERT-based model and BERTScore. However, the standalone GPT-4 method outperformed this approach by a large margin. Ultimately, in our submission, we adopted what we believe to be the optimal prompts and achieved the highest scores in the second batch.

Keywords

Biomedical Question Answer, Large language models (LLMs), Generative Pre-trained Transformer, Zero-shot

1. Introduction

BioASQ[1] has been organizing annual challenges in biomedical semantic indexing and question answering since 2013. This year, BioASQ Task 11b Phase B (QA task)[2] provides biomedical questions along with relevant snippets, and participants are required to generate either the exact answer or the ideal answer using these snippets. The training set for Task 11b Phase B consisted of 4,719 questions, including the test set from the previous year with gold annotations. In addition, it included 330 new test questions for evaluation. The questions were divided into four batches, with 75, 75, 90, 90 questions respectively. A team of biomedical experts from across Europe constructed all the questions and answers. The questions were categorized into four types: Yes/no, factoid, list, and summary. Three types of questions required exact answers: yes/no, factoid, and list. Participants were expected to submit the ideal answer to each question. In Task 11b, each participant could submit up to five results per batch.

Figure 1 illustrates four examples of QA types for BioASQ Task 11b Phase B (QA task). Each instance of BioASQ QA consists of a question and PubMed abstract snippets relevant to the

CLEF 2023: Conference and Labs of the Evaluation Forum, September 18–21, 2023, Thessaloniki, Greece
*Corresponding author.
qazqwe0922@gmail.com (C. Hsueh); phoenix000.taipei@gmail.com (Y. Zhang); ywbonnie@g.ncu.edu.tw (Y. Lu); joyhan@cc.ncu.edu.tw (J. Han); wilailack.meesawad@g.ncu.edu.tw (W. Meesawad); thtsai@csie.ncu.edu.tw (R. T. Tsai)

question. Thus, we framed the task as a query-based multi-document extraction (for the exact answer) and summarization (for the ideal answer). In the previous year, we achieved the highest result in generating ideal answers by using the BioBERT model in combination with linear regression [3].

This year, we observed GPT-4's comprehension capabilities in the field of biology and its advantages in answer generation. We therefore used GPT-4 for answer generation. Specifically, in each batch we developed three or more systems. Particularly in System 1 and System 2, we investigated the impact of prompts on answer generation. We employed Prompt Engineering techniques to select the most suitable prompt. Both systems shared the same prompt, with the only difference being that System 1 utilized GPT-3.5 while System 2 utilized GPT-4. As for System 3, based on the results from the previous year's competition, we found that our research from last year performed well in generating ideal answers. Therefore, we improved upon the previous year's model and utilized its ideal answer for response generation. We relied on System 2's answers for exact answers.

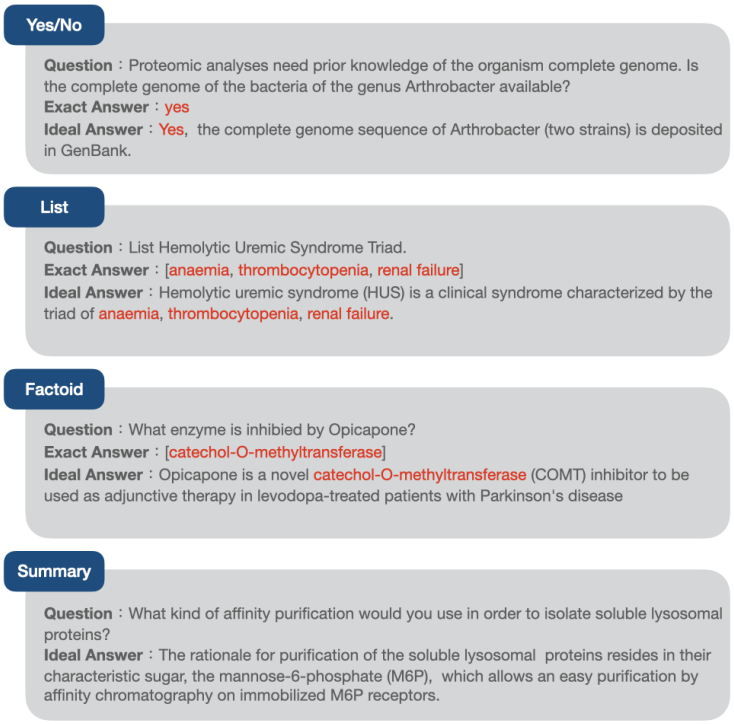

Figure 1: Examples of QA types for BioASQ Task 11b Phase B

2. Related Work

Biomedical knowledge is often acquired by reading academic papers. This process is time-consuming and labor-intensive, and it requires a high level of professional expertise. Biomedi-

cal professionals cannot quickly obtain the required knowledge in a short period of time. The general public is also unable to acquire biomedical knowledge without expert assistance. QA in natural language processing tasks has the potential to solve these problems by providing direct answers to users' questions. This tests machine learning systems' ability to semantically understand, retrieve, and generate answers from existing text. Many QA models based on deep learning have been developed and applied in the past [4].

Well-trained Large language models: Well-trained large language models have emerged as a powerful tool in natural language processing (NLP) tasks, particularly in question answering (QA). These language models, such as GPT-3 and GPT-4, are trained on vast amounts of text data and can understand and generate human-like responses.

In NLP QA tasks, well-trained large language models have shown remarkable performance, surpassing traditional methods and achieving state-of-the-art results. These models excel in comprehending complex questions and generating accurate and contextually relevant answers. They leverage their vast knowledge base to provide detailed explanations, supporting evidence, and even generate creative responses.

According to the GPT-4 Technical Report[5], GPT-4 demonstrates a high level of understanding in the medical domain. This is evidenced by its 75% score on the Medical Knowledge Self-Assessment Program test. Additionally, it obtained an outstanding score of 5 in the AP Biology Exam, a feat accomplished by only 15% of the test takers. This indicates its strong performance in biology-related questions. Therefore, we anticipate that GPT-4 will also deliver favorable results in the BioASQ 11b Phase B task.

3. Method

3.1. Systems

We use different systems in different batch. The detailed configuration of each system can be seen in Table 1

3.2. Dataset

This year's competition provided 4,719 training data samples. Among them, there were 1,130 samples of the summary type, 1,417 samples of the factoid type, 901 samples of the list type, and 1,271 samples of the yes/no type. On average, each question consisted of 12 snippets, with an average length of 203 characters per snippet.

Considering the token limit imposed by OpenAI's API, we extracted only partial information from the snippets. Specifically, in the first two batches, we selected the first 5 snippets and truncated any excessively long sentences to 250 characters. In the subsequent two batches, we input all the snippets. However, for snippets exceeding 250/300 characters, we utilized the ChatCompletions API to perform summarization tasks. This ensured that sentence lengths remained within 250/300 characters.

Table 1
All submitted systems' settings. BioBert's Model field represents that Exact Answer uses GPT-4 results, while Ideal Answer utilizes last year's method. In the Snippet Strategy field, split means to truncate the snippet directly, while summary means to summarize using the same model.

Batch	System Name	Model	Snippet Length	Snippet Strategy
Batch-1	IISR-1	GPT-3	250	split
	IISR-2	GPT-4	250	split
	IISR-3	BioBert	250	split
Batch-2	IISR-1	GPT-3	250	split
	IISR-2	GPT-4	250	split
	IISR-3	BioBert	250	split
Batch-3	IISR-1	GPT-3	250	summary
	IISR-2	GPT-4	250	summary
	IISR-3	BioBert	250	summary
Batch-4	IISR-1	GPT-3	250	summary
	IISR-2	GPT-4	250	summary
	IISR-3	BioBert	250	summary
	IISR-4	GPT-3	300	summary
	IISR-5	GPT-4	300	summary

3.3. Prompting

OpenAI's ChatCompletions API adheres to a predefined format, requiring specific fields for each message. In addition to the "text" field, the "role" field must be configured, which can be categorized as "system", "assistant", or "user".

- The system message: As an optional component, configures the assistant's behavior. It can alter the assistant's personality or furnish explicit instructions regarding its conduct throughout the conversation.
- The user message: Convey requests or comments that require responses from the assistant.
- The assistant message: Retain prior assistant responses, while also allowing developers to compose them as illustrative instances of desired behavior.

Snippets: We observed that ChatGPT incorporates past responses, and we aim to leverage this feature to achieve a similar effect of having ChatGPT read through snippets before answering questions. Therefore, for the snippets, we adopt the format of assistant messages, separating snippets from questions, and directly appending them before the questions. We do not include any additional prompts.

Questions: We experimented with various prompts to guide ChatGPT in generating the desired responses. Ultimately, we opted for a direct approach where ChatGPT generates responses in a fixed JSON format. This decision was driven by our observation that the Exact Answer and Ideal Answer often have a certain degree of overlap. By combining both questions in a single prompt, we encouraged ChatGPT to avoid generating completely unrelated responses. Additionally, imposing a fixed response format greatly improved data processing ef-

ficiency. Across the four batches, we employed similar prompts without significant variations. Please refer to Table 2 for the details of the relevant prompts.

Table 2
The prompts using on ChatGPT

Question Types or Tasks	Prompts
Summary	Reply to the answer clearly and easily in less than 3 sentences. The first question is:{QUESTION_BODY}
Yes/No	You can only use JSON format to answer my questions. The format must be {"exact_answer":"", "ideal_answer":""}, where exact_answer should be "yes" or "no", and ideal_answer is a short conversational response starting with yes/no then follow on the explain. The first question is:{QUESTION_BODY}
List	You can only use JSON format to answer my questions. The format must be {"exact_answer":[], "ideal_answer":""}, where exact_answer is a list of precise key entities to answer the question, and ideal_answer is a short conversational response containing an explanation. The first question is:{QUESTION_BODY}
Factoid	You can only use JSON format to answer my questions. The format must be {"exact_answer":[], "ideal_answer":""}. where exact_answer is a list of precise key entities to answer the question. ideal_answer is a short conversational response containing an explanation. The first question is:{QUESTION_BODY}
To summarize the snippets	Conclusion and summarize this context in less than {MAX_SNIPPET_LEN} letters: {SNIPPET_BODY}

3.4. Strategy

In terms of prompt engineering, we can incorporate certain cues or guidelines, in accordance with the competition rules, to enhance the effectiveness of the responses. The following are some of the strategies we have employed:

- In yes/no type questions, we restrict ChatGPT to only provide responses of 'yes' or 'no.' This approach ensures ChatGPT avoids ambiguous answers.
- We enable ChatGPT to simultaneously respond to both an exact answer and an ideal answer. This approach prevents situations where there are starkly different responses between the two question-answer pairs. Additionally, simultaneous responses encourage ChatGPT to explain its exact answer within the ideal answer. While we do not have explicit experimental evidence, we believe that, similar to the concept of Chain-of-Thought[6], having the language model explain its own answers can enhance the accuracy of the responses.
- When presenting JSON format, we use quotation marks and square brackets to represent strings and lists, respectively. We also provide additional textual descriptions to help ChatGPT understand the expected type of answer it should provide.
- We have observed that the length of a code snippet can impact the length of the generated response. Therefore, in summary-type questions, we limit ChatGPT to providing

answers in three sentences. This implicitly avoids excessively long responses without explicitly specifying a specific word count. This approach helps prevent ChatGPT artificially elongating short answers to the question or generating extremely long responses.

When formulating prompts, we intentionally avoid defining rules or restrictions in excessive detail or complexity. Doing so could potentially result in responses lacking diversity. Therefore, we leave some room for ChatGPT to explore and generate more varied answers, allowing creativity within certain boundaries.

3.5. Procedure

Prompt engineering is an experimental and iterative process that requires continuous experimentation, evaluation, and improvement. Depending on the specific task and dataset, different steps and combinations of methods may be necessary. The key is to adjust and optimize based on actual circumstances to achieve the best model outputs. In our experiment, we followed the following steps:

1. Definition: Confirm the specific task objectives and define the model's input and output.
2. Analysis: Analyze the characteristics and specifications of the dataset.
3. Design: Design a prompt that combines different strategies.
4. Evaluation: Examine and analyze the output results.
5. Optimization: Attempt to optimize the strategies and explore combinations of different methods.
6. Iteration: Repeat steps 3 to 5 continuously until satisfactory output results are achieved.

Table 3
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			FIN
		Acc	maF1	SAcc	LAcc	MRR	Precision	Recall	F1	
Batch-1	IISR-1	0.917	0.906	0.421	0.421	0.421	0.719	0.667	0.684	0.674
	IISR-2	0.958	0.952	0.526	0.526	0.526	0.642	0.570	0.597	0.694
	IISR-3	0.708	0.415	-	-	-	-	-	-	-
Batch-2	IISR-1	1.000	1.000	0.500	0.546	0.523	0.486	0.331	0.368	0.630
	IISR-2	1.000	1.000	0.546	0.636	0.591	0.510	0.358	0.398	**0.663**
	IISR-3	0.583	0.368	0.546	0.636	0.591	0.510	0.358	0.398	0.524
Batch-3	IISR-1	0.917	0.906	0.423	0.423	0.423	0.515	0.458	0.459	0.600
	IISR-2	0.917	0.911	0.385	0.423	0.404	0.652	0.606	0.605	0.642
	IISR-3	0.625	0.384	0.385	0.423	0.404	0.652	0.606	0.605	0.554
Batch-4	IISR-1	1.000	1.000	0.387	0.419	0.403	0.717	0.648	0.671	0.691
	IISR-2	0.929	0.918	0.419	0.419	0.419	0.640	0.662	0.636	0.661
	IISR-3	0.286	0.222	0.419	0.419	0.419	0.242	0.184	0.197	0.301
	IISR-4	1.000	1.000	0.419	0.419	0.419	0.667	0.575	0.596	0.672
	IISR-5	0.929	0.918	0.452	0.452	0.452	0.713	0.684	0.681	0.687

Table 4
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	0.378	0.314	0.361	0.290
	IISR-2	0.415	0.329	0.403	0.309
	IISR-3	0.448	0.406	0.439	0.396
Batch-2	IISR-1	0.339	0.287	0.340	0.282
	IISR-2	0.355	0.301	0.352	0.290
	IISR-3	0.339	0.306	0.336	0.295
Batch-3	IISR-1	0.381	0.345	0.383	0.342
	IISR-2	0.378	0.323	0.378	0.317
	IISR-3	0.376	0.331	0.364	0.314
Batch-4	IISR-1	0.350	0.342	0.345	0.331
	IISR-2	0.331	0.300	0.332	0.293
	IISR-3	0.333	0.340	0.314	0.317
	IISR-4	0.345	0.336	0.339	0.326
	IISR-5	0.345	0.321	0.341	0.309

4. Result

The final scores can be obtained from the BioASQ competition results page. These scores are categorized into Exact Answer (Table 3) and Ideal Answer (Table 4). In the Exact Answer category, we included an additional FIN Score, which utilizes the same final ranking score calculation method as the previous year. Although we do not yet have access to the Manual Scores in the Ideal Answer category, we found that the IISR-2 system in Batch 2 achieved the highest score in the FIN metric within the Exact Answer category. This suggests that if the final ranking score calculation remains the same as last year, we would secure the first position in Batch 2 Exact Answer.

5. Discussion And Conclusions

In this year's competition, we observed widespread use of Generative Transformers. However, training a Generative Transformer model effectively is often challenging in typical scenarios. Therefore, we heavily rely on pre-trained large-scale language models that already demonstrate a certain level of generality. Our results in this competition reflect this observation, as the GPT model far exceeded our fine-tuned BioBert model.

When most participants utilize OpenAI's API to generate results, it becomes crucial to guide ChatGPT in providing the expected answers. Specifically, the most critical aspect is how to provide key prompts without exceeding the token limit.

Our high performance in Batch 2 indirectly indicates our strategies' effectiveness. While it is difficult to precisely analyze which strategy contributed the most to the improvement in performance, the summarized explanation of our strategies includes: 1) Using the Assistant role to directly incorporate snippets, 2) Simultaneously addressing both Exact Answer and Ideal

Answer, 3) Having ChatGPT respond in a fixed JSON format, and 4) Summarizing excessively long snippets before processing.

Despite our efforts, we have observed that some other teams performed better in this competition. Therefore, we have been reflecting on why there were disparities in performance despite using the same model. We believe that there are still many areas for improvement. For example, we can employ more scientific methods to determine which snippets should be referenced, conduct more rigorous validation and evaluation of experimental results, and even explore whether to use simple English words or include subject pronouns in the prompts.

By continuously seeking ways to enhance our approach, we aim to bridge the performance gap and achieve better results in future iterations.

References

[1] A. Nentidis, G. Katsimpras, A. Krithara, S. Lima-López, E. Farré-Maduell, L. Gasco, M. Krallinger, G. Paliouras, Overview of bioasq 2023: The eleventh bioasq challenge on large-scale biomedical semantic indexing and question answering, in: Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Fourteenth International Conference of the CLEF Association (CLEF 2023), 2023.

[2] A. Nentidis, G. Katsimpras, A. Krithara, G. Paliouras, Overview of bioasq tasks 11b and synergy11 in clef2023, in: Working Notes of CLEF 2023 - Conference and Labs of the Evaluation Forum, 2023.

[3] H.-H. Ting, Y. Zhang, J.-C. Han, R. T.-H. Tsai, Ncu-iisr/as-gis: Using bertscore and snippet score to improve the performance of pretrained language model in bioasq 10b phase b, CEUR Workshop Proceedings 3180 (2022).

[4] T. Möller, A. Reina, R. Jayakumar, M. Pietsch, COVID-QA: A question answering dataset for COVID-19, in: Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020, Association for Computational Linguistics, Online, 2020. URL: https://aclanthology.org/2020.nlpcovid19-acl.18.

[5] OpenAI, Gpt-4 technical report, 2023. arXiv:2303.08774.

[6] 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.