Is ChatGPT a Biomedical Expert?

Exploring the Zero-Shot Performance of Current GPT Models in Biomedical Tasks

Samy Ateia[1], Udo Kruschwitz[1]

[1]*Information Science, University of Regensburg, Regensburg, Germany*

Abstract

We assessed the performance of commercial Large Language Models (LLMs) GPT-3.5-Turbo and GPT-4 on tasks from the 2023 BioASQ challenge. In Task 11b Phase B, which is focused on answer generation, both models demonstrated competitive abilities with leading systems. Remarkably, they achieved this with simple zero-shot learning, grounded with relevant snippets. Even without relevant snippets, their performance was decent, though not on par with the best systems. Interestingly, the older and cheaper GPT-3.5-Turbo system was able to compete with GPT-4 in the grounded Q&A setting on Factoid and List answers. In Task 11b Phase A, focusing on retrieval, query expansion through zero-shot learning improved performance, but the models fell short compared to other systems. The code needed to rerun these experiments is available through GitHub[1].

Keywords

Zero-Shot Learning, LLMs, BioASQ, GPT-4, NER, Question Answering

1. Introduction

Recently released ChatGPT models GPT-3.5-Turbo and GPT-4 [1] and their unprecedented zero-shot performance in a variety of tasks, sparked a surge in the development and application of LLMs. By participating in the eleventh CLEF BioASQ challenge [2], we wanted to explore how well these systems perform in specialized domains and whether they can compete with expert fine-tuned systems.

1.1. BioASQ Challenge

BioASQ is a series of large-scale biomedical challenges associated with the CLEF 2023 conference. Its 11th iteration comprises three tasks [2]:

1. Synergy On Biomedical Semantic QA For Developing Issues
2. Biomedical Semantic QA
3. MedProcNER On MEDical PROCedure Named Entity Recognition

This paper focuses on the second and third tasks, the two tasks we participated in. The Biomedical Semantic QA task (Task B) is subdivided into Phase A (document retrieval and snippet extraction) and Phase B (Question Answering) [3].

[1]https://github.com/SamyAteia/bioasq

CLEF 2023: Conference and Labs of the Evaluation Forum, September 18–21, 2023, Thessaloniki, Greece

Samy.Ateia@stud.uni-regensburg.de (S. Ateia); udo.kruschwitz@ur.de (U. Kruschwitz)

We will start with a brief overview of some related work in Section 2 before outlining the experimental setup in Section 3. Section 4 presents our methodology followed by a discussion of Results in Section 5. Finally, we will also touch on ethical issues (Section 6) and offer some conclusions (Section 7).

2. Related Work

To motivate our approach and contextualise our contribution we will briefly discuss related work on recently released generative pre-trained transformer models, have a look at few-shot and zero-shot learning and touch on professional search, i.e. search in a professional context.

2.1. GPT Models

Recently released generative pre-trained transformer (GPT) models GPT-4 and GPT-3.5-turbo are based on the transformer architecture [4] and pre-trained on the next token prediction task. These models are additionally fine-tuned with reinforcement learning from human feedback, which greatly improves their ability to follow instructions and the perceived utility of their generations [5]. OpenAI states that GPT-3.5-turbo is additionally optimized for chats, but does not disclose the exact training procedure used[1].

GPT-4 is the most recent and best performing model of OpenAI, which is, as of this writing, only programmatically accessible through closed beta API access[2]. It exhibits human-level performance on various professional and academic benchmarks and can process images as well as text [1].

2.2. Few and Zero-Shot Learning

These models improve over the earlier GPT-3 model which showed that in certain tasks sufficiently big LLMs can compete with fine-tuned transformer models using only few-shot learning, which greatly reduces the need for extensive training data [6].

In the **few-shot learning** setting, the GPT models are prompted with a text that contains a few examples of the tasks at hand, for example, multiple question-answer pairs, and at the end only the current question for which an answer should be generated by the model. The model then ideally completes this text by writing the correct answer.

In the **zero-shot learning** setting, the model is not supplied with any examples but rather only a direct question or abstract task description and is ideally able to generate a useful completion that answers the question or solves the task [7].

Zero-shot and few-shot learning is especially interesting for applications in specialized domains with no or sparse training data available. Prior work in the biomedical domain has shown that language models pre-trained on in-domain data outperform models pre-trained on open domain data [8][9][10]. In this work, we want to explore whether these new GPT models, that are extensively trained on vast amounts of open domain data, can compete with specialized fine-tuned models that are expected to participate in the challenge.

[1] https://platform.openai.com/docs/model-index-for-researchers
[2] https://openai.com/product/gpt-4

Even though these models are proprietary and neither the architecture nor the specific training process is known, several open-source alternatives have been developed such as OPT [11], BLOOM [12], or Pythia [13]. Projects based on these and other open source models are constantly improving, and some are already nearly reaching GPT-3.5-turbo level performance [14]. We therefore believe that studying these commercial models is valuable for establishing a baseline in zero-shot performance for upcoming open-source alternatives. These alternatives could potentially challenge state-of-the-art (SOTA) systems across a wide range of natural language processing (NLP) tasks.

2.3. Professional Search

Professional search is search conducted in a work context [15]. This is an everyday activity for many professionals that comes with specific requirements which are different from the requirements of generic Web search [16]. The BioASQ challenge can be framed as a form of professional search in which the searchers are biomedical experts aiming to find answers to domain-specific questions.

Automatic query expansion plays a key part in many professional search contexts including search by healthcare information professionals, patent agents and recruitment professionals [17] as well as in conducting systematic reviews [18]. What is ultimately being submitted to the search system can turn out to be a fairly complex search strategy, a query involving domain-specific information based around Boolean operators. This is one of the motivations for us to explore automatic query expansion in our methodology.

3. Experimental Setup

We describe the experimental setup of the two BioASQ tasks that we participated in, Task 11 B and MedProcNER. For Task 11 B a benchmark dataset with training and test biomedical questions in English along with reference answers was used that has been created based on questions by biomedical experts [19].

3.1. Task 11 B: Biomedical Semantic QA

For **Phase A**, the participating systems receive a list of biomedical questions such as "*Which protein is targeted by Herceptin?*" and should retrieve a list of up to 10 most relevant articles from the PubMed Annual Baseline Repository for 2023[3]. Additionally, the systems should also create a list of at most 10 most relevant snippets extracted from the previously retrieved article titles or abstracts. Participating systems are compared based on the Mean Average Precision (MAP) metric.

In **Phase B**, the participating systems receive the same questions as in Phase A, along with a set of gold (correct) articles and snippets selected by biomedical experts. They should then generate an *ideal* paragraph sized (at most 200 words) answer based on these snippets. The questions are also tagged with either *Yes/No*, *Factoid*, *Summary*, or *List* type indicating the format for an additional *exact* answer that should be created by these systems.

[3]https://lhncbc.nlm.nih.gov/ii/information/MBR.html

- *Yes/no* questions require the exact answer to be either "yes" or "no".
- *Factoid* question require the exact answer to be a list of up to 5 entity names or other short expressions ordered by decreasing confidence.
- *List* questions require the exact answer to be a list of up to 100 entity names or similar short expressions.
- *Summary* questions do not require an additional exact answer, only the *ideal* answer needs to be returned.

3.2. MedProcNER: MEDical PROCedure Named Entity Recognition

The **MedProcNER** task [20] focuses on the detection and mapping of medical procedures in Spanish texts. It consists of three subtasks:

- In subtask 1, systems have to identify medical procedures from Spanish clinical reports.
- In subtask 2, systems have to map the medical procedures identified in subtask 1 to SNOMED CT codes [21].
- In subtask 3, systems have to assign SNOMED CT codes to the full clinical report for later indexing.

4. Methodology

4.1. Model

We accessed GPT-3.5-turbo and GPT-4 through the OpenAI API[4]. We used a simple system message to set the behavior of the model, which can be seen in Listing 1.

Listing 1: System Message

```
You are BioASQ–GPT, an AI expert in question answering, research, and information
retrieval in the biomedical domain.
```

This system message was then followed by task specific zero-shot prompts, including necessary information such as the questions, snippets, or retrieved article titles. More details on these prompts can be found in the subsection corresponding to the particular task. Prompt engineering has developed into a very active field and at this point we should note that there is scope for plenty of future work exploring more systematically the best way of prompting the system for the task at hand.

We experimented with a subset of the BioASQ training and development data to explore the system's behavior and evaluate the performance of individual modules.

Additional parameters that were sent in the API request to the models were *temperature* which controls the randomness of completion; *frequency_penalty* which discourages repetition of words or phrases; and *presence_penalty* which has a similar effect. We set *temperature* to 0 for all requests to have reproducible results over multiple runs.

[4]https://platform.openai.com/docs/guides/chat/introduction

As these models are currently **non-deterministic**, even with temperature set to 0, there is a residual randomness in the generations, which can lead to slightly different results in each run[5]. We also conducted a limited test to roughly estimate the variance of the results by repeating five runs over the same 50 questions.

4.2. Task 11 B

4.2.1. Phase A

Our approach used zero-shot learning for query expansion, query reformulation and reranking directly with the models. For document retrieval, we queried the eUtils API with a *maxdate* cutoff corresponding to the creation date of the relevant 2023 PubMed snapshot. The Entrez Programming Utilities (eUtils) API is a set of web applications provided by the National Center for Biotechnology Information (NCBI), which offers programmatic access to the various databases and functionalities of the NCBI resources, such as PubMed. We also used the sort by relevance option of PubMed and retrieved only the top 50 results for a given query.

We acknowledge that querying the live PubMed database with the corresponding date cutoff is not the same as searching through the downloaded static snapshot or using the search interface provided by the BioASQ organizers. Articles could be deleted or modified in PubMed, which could affect the reproducibility and comparability of the results with other systems. To estimate the impact of this approach, we looked up all articles that were included in the gold set provided in Phase B of the task after the challenge concluded and found that one out of the 899 referenced articles was no longer retrievable in PubMed[6].

We were most interested in the impact of the query expansion step and therefore conducted one run with and one without query expansion for both models, where we instead sent the question directly as a query to PubMed.

The exact steps were:

1. Query expansion
2. Search on PubMed
3. Query refinement only if no documents were found and one additional search on PubMed
4. Reranking of top 50 articles based on title string

All of these steps were executed automatically in Python without manual intervention, the exact code used is available on GitHub[7]. The zero-shot learning prompt used for query expansion can be seen in Listing 2. Where the placeholder *{question}* was replaced by the question that was currently processed by the system. For query expansion, we set *frequency_penalty* to 0.5 and *presence_penalty* to 0.1.

Some example query expansions for this prompt can be seen in Listing 3. Interestingly, these models seem to not only know what Boolean syntax is accepted by PubMed but also important

[5]https://platform.openai.com/docs/models/gpt-3-5
[6]Article from batch 4 that is no longer accessible in PubMed: http://www.ncbi.nlm.nih.gov/pubmed/36459075
[7]https://github.com/SamyAteia/bioasq

Listing 2: Query Expansion Prompt

```
{"role": "user", "content": f"""Expand this search query:
'{question}' for PubMed by incorporating synonyms and additional terms that closely relate to
the main topic and help reduce ambiguity. Assume that phrases are not stemmed; therefore,
generate useful variations. Return only the query that can directly be used without any
explanation text. Focus on maintaining the query's precision and relevance to the original
question."""}
```

internal fields such as *MeSH Terms* and the syntax on how to query on these fields, but these were not often used in the expanded queries.[8]

Listing 3: Query Expansion Example

```
Question: What are the outcomes of ubiquitination?

Expanded Query: ("ubiquitination" OR "ubiquitin modification" OR "ubiquitin conjugation"
OR "ubiquitin pathway") AND ("outcomes" OR "effects" OR "consequences")

Question: What is the incidence of Leigh syndrome?

Expanded Query: ("Leigh syndrome"[MeSH Terms] OR "Leigh syndrome"[All Fields] OR "
subacute necrotizing encephalomyelopathy"[All Fields]) AND ("incidence"[MeSH Terms] OR
"incidence"[All Fields] OR "prevalence"[MeSH Terms] OR "prevalence"[All Fields])
```

For the optional query reformulation step, we used the prompt in Listing 4. This step was introduced after it became clear that some queries constructed by the models were overly specific and returned no results. The placeholder *{question}* in the prompt was replaced by the question that was currently processed by the system, and the placeholder *{original_query}* was replaced by the original expanded query that returned no results. For query reformulation, we set *frequency_penalty* to 0.6 and *presence_penalty* to 0.2. An example of a query reformulation that generated a slightly broader query that then led to some results can be seen in Listing 5. Additionally, terms added to the query are highlighted in gray.

Listing 4: Query Reformulation Prompt

```
{"role": "user", "content": f"""Given that the following search query for PubMed has returned
no documents, please generate a broader query that retains the original question's context
and relevance. Assume that phrases are not stemmed; therefore, generate useful variations.
Return only the query that can directly be used without any explanation text. Focus on
maintaining the query's precision and relevance to the original question. Original question: '{
question}', Original query: '{original_query}'."""}
```

[8]The identification of suitable MeSH terms in structured queries for systematic reviews has been explored in detail elsewhere, e.g. [22, 23]

Listing 5: Query Reformulation Example

```
Question: Can skin picking phenotype present following methylphenidate treatment?
Query: ("skin picking" OR "excoriation disorder" OR "dermatillomania" OR "compulsive skin
picking") AND (phenotype OR presentation OR manifestation) AND ("methylphenidate
treatment" OR "methylphenidate therapy" OR "methylphenidate administration")

Reformulated Query: ("skin picking" OR "excoriation disorder" OR "dermatillomania" OR "
compulsive skin picking") AND (phenotype OR presentation OR manifestation OR
 symptoms ) AND ("methylphenidate treatment" OR "methylphenidate therapy" OR "
methylphenidate administration" OR  "methylphenidate use" )
```

For the final reranking step, we took the titles of the top 50 returned articles as returned by the relevancy sort from PubMed and prompted the model to rerank these articles given the original question and return the top 10 articles. The prompt used for the reranking can be seen in Listing 6, where *{articles_str}* is replaced by the list of returned article titles, *{question}* is replaced by the question that is currently processed by the system, and *{nr_of_articles}* is replaced by the 10 or fewer articles if less relevant articles were returned by PubMed. For reranking, we set *frequency_penalty* to 0.3 and *presence_penalty* to 0.1.

Listing 6: Reranking Prompt

```
{"role": "user", "content": f"{articles_str} \n\n Given these articles and the question: '{question
}'. Rerank the articles based on their relevance to the question and return the top {
nr_of_articles} most relevant articles as a comma separated list of their index ids. Don't
explain your answer, return only this list, for example: '1, 2, 3, 4' "}
```

The returned list was then mapped back to the articles retrieved from PubMed, and these were returned as the required output of Phase A.

We also explored the extraction of snippets for the Phase A task but abandoned it, as it required sending all abstracts of the 10 returned papers for processing to the model, which was especially expensive for the GPT-4 model because API usage is priced on token counts, and we were exploring these models on a limited budget.

4.2.2. Phase B

In Phase B, we used the gold (correct) snippets from the test set and sent them along with the question and description of the answer format to the model.

We also conducted a test where this grounding information in the form of relevant snippets was omitted and just the question and description of the answer format were sent to the models.

The prompts utilized for generating these answer types are listed as follows: for ideal answers, refer to Listing 7; for Yes/No answers, see Listing 8; for List answers, Listing 9; and for Factoid responses, see Listing 10.

Listing 7: Ideal Answer Prompt

```
{"role": "user", "content": f""" {snippets}\n\n\ '{question['body']}'. Answer this question by returning a single paragraph–sized text ideally summarizing the most relevant information. The maximum allowed length of the answer is 200 words. The returned answer is intended to approximate a short text that a biomedical expert would write to answer the corresponding question (e.g., including prominent supportive information)."""}
```

Listing 8: Yes/No Answer Prompt

```
{"role": "user", "content": f" {snippets}\n\n\ '{question['body']}'. You *must answer* only with lowercase 'yes' or 'no' even if you are not sure about the answer."}
```

Listing 9: Factoid Answer Prompt

```
{"role": "user", "content": f" {snippets}\n\n\ '{question['body']}'. Answer this question by returning only a JSON string array of entity names, numbers, or similar short expressions that are an answer to the question, ordered by decreasing confidence. The array should contain at max 5 elements but can contain less. If you don't know any answer return an empty list. Return only this list, it must not contain phrases and **must be valid JSON**."}
```

Listing 10: List Answer Prompt

```
{"role": "user", "content": f" {snippets}\n\n\ '{question['body']}'. Answer this question by only returning a JSON string array of entity names, numbers, or similar short expressions that are an answer to the question (e.g., the most common symptoms of a disease). The returned list will have to contain no more than 100 entries of no more than 100 characters each. If you don't know any answer return an empty list. Return only this list, it must not contain phrases and **must be valid JSON**."}
```

In all these prompts, *{question['body']}* is replaced by the question that is currently processed by the system, and *{snippets}* is replaced by the snippets provided by the test set.

For all answer types, we set *frequency_penalty* to 0.5. *Presence_penalty* was set to 0.3 for Yes/No answers, to 0.1 for both List and Factoid answers, and to 0.7 for the ideal answer type.

4.3. MedProcNER

For the MedProcNER task, we translated all prompt templates, including the system prompt, to Spanish using and comparing deepL[9]) and ChatGPT[10]. For substask 1, instead of using zero-shot

[9]https://www.deepl.com/en/translator
[10]https://chat.openai.com/

Listing 11: MedProcNER Prompt

```
conversation = [{'role': 'system', 'content': """Eres un asistente útil que extrae procedimientos
 médicos de textos médicos en español. Un procedimiento médico se refiere a cualquier acció
n diagnóstica, terapéutica, médica o quirúrgica realizada en un paciente. Tu respuesta debe
ser una lista de procedimientos en formato JSON válido."""}]
for input, output in examples:
    conversation.append({'role': 'user', 'content': f'{input}'})
    conversation.append({'role': 'assistant', 'content': json.dumps(output)})
conversation.append({'role': 'user', 'content': f"""Extraiga todos los procedimientos médicos
del texto delimitado por tres comillas invertidas. Devuelve una lista vacía si no se menciona
ninguno. {text}"""})
```

prompting as before, we instead explored the few-shot prompting approach, where we included three examples from the training set into the request sent to the OpenAI API. We also compared the performance of GPT-3.5-turbo and GPT-4.

The relevant Python code part that constructed the prompt can be seen in Listing 11. The *examples* list mentioned therein contained three examples taken from the training set.

For substask 2, we used the gazetteer file provided by the MedProcNER task organizers. We filtered the file for all SNOMED CT codes that were tagged as procedure and stemmed their terms, and used Levenshtein distance based fuzzy matching to find an entry for a procedure. The detailed code used for all tasks is available on the aforementioned GitHub repository.

For subtask 3, we just joined all SNOMED CT codes identified in subtask 2 for one document.

5. Results

The systems participating in the Biomedical Semantic Q&A task were evaluated in four batches. Results are reported for every batch. For readability, we only included the results of our systems and the top performing systems. The full result tables are publicly available on the BioASQ website[11]

5.1. Task 11 B Phase A

We participated with 4 systems in Task 11 B Phase A, the systems' names and their properties are listed as follows:

- UR-gpt4-zero-ret corresponds to GPT-4 with query expansion.
- UR-gpt3.5-turbo-zero corresponds to GPT-3.5-turbo with query expansion.
- UR-gpt4-simple corresponds to GTP-4 without query expansion.
- UR-gpt3.5-t-simple corresponds to GPT-3.5-turbo without query expansion.

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

The following Table 1 shows the results of our systems participating in the 4 batches. MAP was the official metric to compare the systems. N stands for the number of participating systems in each batch.

Table 1
Task 11 B Phase A, Batches 1-4

Batch	Position	System	Precision	Recall	F-Measure	**MAP**	GMAP
Batch 1 N = 33	1	Top Competitor	0.2118	0.6047	0.2774	0.4590	0.0267
	19	UR-gpt4-zero-ret	0.1664	0.3352	0.1955	0.2657	0.0009
	21	UR-gpt3.5-turbo-zero	0.1488	0.2847	0.1782	0.2145	0.0009
	24	UR-gpt4-simple	0.1654	0.2508	0.1799	0.1809	0.0005
	25	UR-gpt3.5-t-simple	0.1600	0.2290	0.1734	0.1769	0.0003
Batch 2 N = 33	1	Top Competitor	0.1027	0.5149	0.1618	0.3852	0.0104
	20	UR-gpt4-simple	0.0945	0.3011	0.1277	0.1905	0.0011
	21	UR-gpt3.5-turbo-zero	0.1153	0.2977	0.1455	0.1736	0.0008
Batch 3 N = 35	1	Top Competitor	0.0800	0.4776	0.1320	0.3185	0.0049
	21	UR-gpt3.5-turbo-zero	0.1295	0.3258	0.1646	0.2048	0.0008
	22	UR-gpt4-zero-ret	0.1086	0.2289	0.1303	0.1930	0.0003
	23	UR-gpt4-simple	0.1089	0.2102	0.1238	0.1727	0.0002
	24	UR-gpt3.5-t-simple	0.1078	0.1981	0.1217	0.1553	0.0002
Batch 4 N = 27	1	Top Competitor	0.0933	0.4292	0.1425	0.3224	0.0030
	18	UR-gpt4-zero-ret	0.0791	0.1728	0.0933	0.1251	0.0002
	19	UR-gpt3.5-turbo-zero	0.0922	0.1956	0.1025	0.1139	0.0002
	20	UR-gpt4-simple	0.0785	0.1563	0.0864	0.1010	0.0002
	21	UR-gpt3.5-t-simple	0.0752	0.1319	0.0810	0.0912	0.0001

One observation is that GPT-4 achieved better results than GPT-3.5-turbo in all batches except batch 3. It seems to perform better in both query expansion and reranking without query expansion. Query expansion consistently improves the results for all models in all batches. It greatly improves recall in all batches, and in most batches, precision is also slightly increased except in batch 1, where it leads to decreased precision for GPT-3.5-turbo but an overall improved F1 score.

In general, our approach performs worse than most systems. This could be due to the fact that we do not do any embedding based neural retrieval, but instead only rely on the keywords created by the models in the query expansion step and the relevancy ranking provided by PubMed. The reranking window of only 50 article titles might also be too small, or the information provided by the titles is not sufficient for a more effective reranking. A thorough ablation study in future work could help explain the contribution of these individual factors to the overall system performance.

Using only query expansion in the retrieval phase and not having to do any embedding calculations during indexing does come with advantages for applying such an approach to existing or huge search use-cases where efficient reindexing with more advanced embedding based approaches might not be feasible. On the other hand, the used models do take several seconds to create results for both reranking and query expansion, which could limit their usefulness in classical enterprise-search use-cases if sub-second response times are expected.

5.2. Task 11 B Phase A

We participated with 4 systems in Task 11 B Phase B, the systems' names and their properties are listed as follows:

- *UR-gpt4-zero-ret* corresponds to GPT-4 grounded with snippets.
- *UR-gpt3.5-turbo-zero* corresponds to GPT-3.5-turbo grounded with snippets.
- *UR-gpt4-simple* corresponds to GTP-4 answering directly without reading snippets.
- *UR-gpt3.5-t-simple* corresponds to GPT-3.5-turbo answering directly without reading snippets.

We were not able to complete all runs in batches 1 and 2, which is why some results are missing. We report the results for each answer format (Yes/No, Factoid, List) separately in the following tables. For readability, we again only included the results of our systems and the top-performing systems, the full result tables are publicly available on the BioASQ website[12].

Table 2
Task 11 B Phase B, Yes/No Questions Batches 1-4

Batch	Position	System	Accuracy	F1 Yes	F1 No	**Macro F1**
Batch1 N = 33	1	Top Competitor	0.9583	0.9697	0.9333	0.9515
	8	UR-gpt4-zero-ret	0.9167	0.9412	0.8571	0.8992
	9	UR-gpt4-simple	0.9167	0.9412	0.8571	0.8992
	13	UR-gpt3.5-turbo-zero	0.8750	0.9091	0.8000	0.8545
Batch2 N = 42	1	Top Competitor	1.0000	1.0000	1.0000	1.0000
	7	UR-gpt4-zero-ret	0.9583	0.9655	0.9474	0.9564
	12	UR-gpt3.5-turbo-zero	0.9167	0.9333	0.8889	0.9111
Batch3 N = 47	1	Top Competitor	1.0000	1.0000	1.0000	1.0000
	9	UR-gpt4-zero-ret	0.9167	0.9375	0.8750	0.9063
	12	UR-gpt4-simple	0.8750	0.9032	0.8235	0.8634
	14	UR-gpt3.5-turbo-zero	0.8750	0.9091	0.8000	0.8545
	21	UR-gpt3.5-t-simple	0.7917	0.8485	0.6667	0.7576
Batch4 N = 52	1	Top Competitor	1.0000	1.0000	1.0000	1.0000
	7	UR-gpt4-zero-ret	0.9286	0.8889	0.9474	0.9181
	14	UR-gpt3.5-turbo-zero	0.9286	0.8571	0.9524	0.9048
	19	UR-gpt4-simple	0.7857	0.7273	0.8235	0.7754
	29	UR-gpt3.5-t-simple	0.4286	0.5000	0.3333	0.4167

In the Yes/No question format, our results indicate that GPT-4 surpasses GPT-3.5-turbo in both the grounded and ungrounded settings. For batches 1 and 3, the ungrounded GPT-4 system *UR-gpt4-simple* even showed a tendency to perform better than the grounded variant of GPT-3.5-turbo *UR-gpt3.5-turbo-zero* as can be seen in Table 2.

In the Factoid question format, both grounded GPT-4 and grounded GPT-3.5-turbo achieved an MRR score of 0.5789 taking first and second place over all other systems. In the remaining batches, GPT-3.5-turbo stayed consistently in the top 6 systems, while GPT-4 only reached 11th and 13th place in batches 3 and 4. This mixed performance comparison between GPT-3.5-turbo and GPT-4 was also observed in the List question format, where GPT-3.5-turbo achieved 1st

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

Table 3
Task 11 B Phase B, Factoid Questions Batches 1-4

Batch	Position	System	Strict Acc.	Lenient Acc.	**MRR**
Batch1 N = 33	**1**	UR-gpt4-zero-ret	0.5789	0.5789	0.5789
	2	UR-gpt3.5-turbo-zero	0.5263	0.6316	0.5789
	3	Next Competitor	0.5263	0.6316	0.5570
	22	UR-gpt4-simple	0.2105	0.2632	0.2368
Batch2 N = 42	1	Top Competitor	0.5455	0.6364	0.5909
	2	Next Competitor	0.5455	0.6364	0.5909
	3	UR-gpt3.5-turbo-zero	0.5455	0.5909	0.5682
	4	UR-gpt4-zero-ret	0.5455	0.5909	0.5682
Batch3 N = 47	1	Top Competitor	0.4615	0.6538	0.5205
	5	UR-gpt3.5-turbo-zero	0.5000	0.5000	0.5000
	11	UR-gpt4-zero-ret	0.4615	0.5000	0.4808
	22	UR-gpt4-simple	0.2692	0.4615	0.3654
	27	UR-gpt3.5-t-simple	0.3077	0.3077	0.3077
Batch4 N = 52	1	Top Competitor	0.6452	0.8710	0.7323
	6	UR-gpt3.5-turbo-zero	0.6452	0.6452	0.6452
	13	UR-gpt4-zero-ret	0.5161	0.6129	0.5645
	30	UR-gpt3.5-t-simple	0.2581	0.2903	0.2742
	33	UR-gpt4-simple	0.2258	0.2581	0.2366

Table 4
Task 11 B Phase B, List Questions Batches 1-4

Batch	Position	System	Strict Acc.	Lenient Acc.	**MRR**
Batch1 N = 33	1	Top Competitor	0.7861	0.6668	0.7027
	2	UR-gpt3.5-turbo-zero	0.6742	0.7249	0.6917
	8	UR-gpt4-zero-ret	0.6472	0.6530	0.6495
	19	UR-gpt4-simple	0.4000	0.4014	0.3939
Batch2 N = 42	**1**	UR-gpt3.5-turbo-zero	0.4598	0.4671	0.4316
	2	Next Competitor	0.5099	0.3577	0.3980
	4	UR-gpt4-zero-ret	0.3742	0.4369	0.3828
Batch3 N = 47	1	Top Competitor	0.6519	0.6058	0.6049
	3	UR-gpt4-zero-ret	0.5518	0.6597	0.5736
	9	UR-gpt3.5-turbo-zero	0.5600	0.5140	0.5101
	24	UR-gpt3.5-t-simple	0.2690	0.2385	0.2333
	25	UR-gpt4-simple	0.2519	0.2343	0.2305
Batch4 N = 52	1	Top Competitor	0.7139	0.8061	0.7440
	2	UR-gpt4-zero-ret	0.6902	0.7818	0.7191
	10	UR-gpt3.5-turbo-zero	0.6090	0.6710	0.6196
	21	UR-gpt4-simple	0.4440	0.4214	0.4127
	26	UR-gpt3.5-t-simple	0.3944	0.3362	0.3470

place in batch 2 but was behind GPT-4 in batches 3 and 4. The results for the Factoid question format are shown in Table 3 and the results for the List question format are shown in Table 4.

While GPT-4 seems to perform consistently better than GPT-3.5-turbo in the Yes/No question format, there is no clear winner in the more extractive Factoid and List formats.

Both models without grounding information from snippets were not able to compete with the top models but were often placed slightly below the average performing systems, which

is still surprisingly good as in this setting the models need to rely only on the open-domain knowledge acquired during training for answering these questions.

5.3. Task MedProcNER

In the MedProcNER task, GPT-4 performed better than GPT-3.5-turbo, but was not able to compete with the best performing system. The results are shown in Table 5. Our simple gazetteer based entity linking and indexing approach performed poorly compared to the top-performing system. At the time of this writing, the performance of other systems involved in the task has not been published yet.

Table 5
Comparison of F1 scores of different systems for NER, Entity Linking, and Indexing tasks.

Task	Top Performing System F1	GPT-3.5-turbo F1	GPT-4 F1
NER	0.7985	0.3002	0.4814
EL	0.5707	0.1264	0.1976
Indexing	0.6242	0.1785	0.2695

Even though the few-shot NER approach did not compete with the top-performing system in the MedProcNER task, it still indicates that GPT-4 can be used for specialized domains in multilingual tasks while only using a minimal amount of training data.

5.4. Discussion and Future Work

The results from our participation in the BioASQ challenge indicate that current commercial GPT models GPT-3.5-turbo and GPT-4 can compete with other presumably fine-tuned leading systems in question answering in the biomedical domain, while only being zero-shot prompted with relevant snippets. Even without relevant snippets, just relying on the biomedical knowledge aquired during their pre-training, these models were performing better than some of the other systems participating in the task.

One big challenge in using zero-shot learning with these GPT models is prompt-engineering. It still seems to be more of an art than a science and requires considerable testing [24]. During system development, it became clear that the expanded queries in Task 11 B Phase A were sometimes too specific and did not return results. We tried to prompt the models to create broader queries that were not using as many phrase terms that are not stemmed in PubMed, but the overall system performance on our development set declined. We therefore experimented with using GPT-4 to come up with a better prompt by supplying it with the original prompt and the 5 worst-performing and 5 best-performing queries. The new prompt actually increased the performance of the system. This self prompt learning might be an interesting approach to investigate further in future work.

Nevertheless, the zero-shot learning approach makes the usage of these models very accessible, as it does not require thorough data preparation, knowledge about classical deep learning techniques, or advanced programming skills.

A prominent problem in these GPT models are so-called hallucinations [25]. These are unsupported or factually wrong statements in the responses. These problems might be espe-

cially observable in the ideal answer setting. In future work, we want to conduct a thorough investigation of the factuality of the ideal answers and especially compare the grounded and ungrounded settings. This could provide error rate estimates that might be useful for generative search systems in specialized domains.

As noted earlier, these commercial models are not completely deterministic, even when the temperature parameter is set to 0. OpenAI states in their documentation:

> "OpenAI models are non-deterministic, meaning that identical inputs can yield different outputs. Setting the temperature parameter to 0 will make the outputs mostly deterministic, but a small amount of variability may remain."[13]

We had concerns about the potential cascading effect of such residual non-determinism, especially in the context of query expansion. To estimate this variability, we performed a limited test by repeating the retrieval task from Task 11 B Phase A over 50 questions taken from the training set five times with the same model. Our test results showed minimal variance across metrics such as MAP, precision, recall, and F-measure, indicating that while variability exists, its impact is currently minimal, with broader investigations pending for future work.

This residual non-determinism in the model output also led to some instability in the system when we fully relied on the model returning the right output format for further processing. For example, in the Yes/No question format, the evaluation system of the BioASQ organizers expects the answers to be all lowercase, either "yes" or "no". The models often returned variants such as "Yes" or "Yes." even if explicitly prompted not to do so. This necessitated an additional normalization post-processing step.

In the MedProcNER task, where we used few-shot learning, it seemed that the examples greatly assisted the model in returning the correct output format. We suspect that giving even just a few examples is a more effective way to guide the models towards the expected output format than explicitly describing the format in a zero-shot learning prompt.

Even if the models were outputting the right format, the overall system was still unstable due to the instability of the OpenAI API. In every run, there were at least 2–3 requests that failed due to internal server errors or the model being overloaded with requests. Thus, retry loops must be incorporated when accessing such external services.

As usage of these models is priced based on token count, some use-cases might not be financially feasible yet. Only running one evaluation batch with GPT-4 can cost around $10 in model usage. At the same time, the GPT-4 model was still much slower in answering requests than GPT-3.5-turbo. These two factors led us to not participate in the snippet generation task, as this task is especially demanding regarding both the amount of tokens to be processed in the prompt and generated as a response. In general, the economic barrier to using these commercial models may hinder some researchers due to the cost of usage. Also, over-reliance on these models might stifle innovation in other research areas.

We also conducted a limited test with grounding the query expansion by suggesting semantically related terms from the word embeddings supplied by the BioASQ organizers, but these terms led to queries that performed worse than just ungrounded ones. We did not investigate this approach thoroughly and leave it open for future work.

[13] https://platform.openai.com/docs/models/gpt-3-5

Some of our results might indicate that the performance gap between presumably smaller (GPT-3.5-turbo) and more complex models (GPT-4) is narrower in the grounded extractive Q&A setting, because GPT-3.5-turbo sometimes performed better than GPT-4 in answering Factoid or List questions in some of the batches. It would be interesting to see how model performance in this setting scales with model size, and to test whether the use of much smaller generative models is feasible. Some related work in other use-cases already showed promising results in this direction [26][27]. This might open up new possibilities for using these models in enterprise search settings where confidential data must remain on-premise [28].

6. Ethical Considerations

The use of large language models like GPT-3.5-Turbo and GPT-4 in biomedical tasks presents several ethical considerations.

First, we must address data privacy. While these models do not retain specific training examples, there is a remote possibility of them generating outputs resembling sensitive data, or sensitive data included in a prompt might be repeated and further processed in downstream tasks. This issue has to be addressed when employing these models in a real world biomedical context.

Second, as these models may produce factually incorrect outputs or "hallucinations" [25], rigorous fact-checking mechanisms must be applied, especially when used in a biomedical context to prevent the spread of harmful misinformation.

Lastly, large language models operate as black-box algorithms, raising issues of interpretability, transparency, and accountability [29].

In conclusion, the potential of large language models in biomedical tasks is significant, but the ethical implications of their deployment need careful attention.

7. Conclusion

We showed that in context learning, both zero- and few-shot, with recent LLMs trained on human feedback can compete with presumably fine-tuned state-of-the-art systems in some domain-specific questions answering tasks. Zero- and few-shot learning can greatly simplify and speed up the development of complex NLP or IR systems, which might be especially useful for research and prototyping. It also opens up the possibility to improve use-cases where fine-tuning is not feasible due to a lack of available training data.

Prompt engineering for these models poses challenges, and grounding the answer generation with the right context information is an interesting problem for current and future generative search systems research. Even though the currently offered GPT models have severe limitations regarding cost of usage, speed, and factuality, we see promising research towards making these types of models more affordable and accessible and improving their overall performance and factuality.

Acknowledgments

We want to thank the organizers of the BioASQ challenge for setting up this challenge and supporting us during our participation. We are also grateful for the feedback and recommendations of the anonymous reviewers.

References

[1] OpenAI, GPT-4 Technical Report, 2023. arXiv:2303.08774.

[2] 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.

[3] 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.

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is All You Need, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17, Curran Associates Inc., Red Hook, NY, USA, 2017, p. 6000–6010.

[5] 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.

[6] B. et al., Language Models Are Few-Shot Learners, in: Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS'20, Curran Associates Inc., Red Hook, NY, USA, 2020.

[7] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, G. Neubig, Pre-Train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing, ACM Comput. Surv. 55 (2023). URL: https://doi.org/10.1145/3560815. doi:10.1145/3560815.

[8] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, J. Kang, BioBERT: a pre-trained biomedical language representation model for biomedical text mining, Bioinformatics 36 (2020) 1234–1240.

[9] I. Beltagy, K. Lo, A. Cohan, SciBERT: A Pretrained Language Model for Scientific Text, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019, pp. 3615–3620.

[10] Y. Gu, R. Tinn, H. Cheng, M. R. Lucas, N. Usuyama, X. Liu, T. Naumann, J. Gao, H. Poon, Domain-Specific Language Model Pretraining for Biomedical Natural Language Processing, ACM Transactions on Computing for Healthcare (HEALTH) 3 (2020) 1 – 23.

[11] S. Zhang, S. Roller, N. Goyal, M. Artetxe, M. Chen, S. Chen, C. Dewan, M. Diab, X. Li, X. V. Lin, T. Mihaylov, M. Ott, S. Shleifer, K. Shuster, D. Simig, P. S. Koura, A. Sridhar,

T. Wang, L. Zettlemoyer, OPT: Open Pre-trained Transformer Language Models, 2022. arXiv:2205.01068.

[12] S. et al., BLOOM: A 176B-Parameter Open-Access Multilingual Language Model, 2023. arXiv:2211.05100.

[13] S. Biderman, H. Schoelkopf, Q. Anthony, H. Bradley, K. O'Brien, E. Hallahan, M. A. Khan, S. Purohit, U. S. Prashanth, E. Raff, A. Skowron, L. Sutawika, O. van der Wal, Pythia: A Suite for Analyzing Large Language Models Across Training and Scaling, 2023. arXiv:2304.01373.

[14] T. Dettmers, A. Pagnoni, A. Holtzman, L. Zettlemoyer, QLoRA: Efficient Finetuning of Quantized LLMs, 2023. arXiv:2305.14314.

[15] J. I. Tait, An Introduction to Professional Search, Springer International Publishing, Cham, 2014, pp. 1–5. URL: https://doi.org/10.1007/978-3-319-12511-4_1. doi:10.1007/978-3-319-12511-4_1.

[16] S. Verberne, J. He, U. Kruschwitz, G. Wiggers, B. Larsen, T. Russell-Rose, A. P. de Vries, First international workshop on professional search, SIGIR Forum 52 (2018) 153–162.

[17] T. Russell-Rose, P. Gooch, U. Kruschwitz, Interactive query expansion for professional search applications, Business Information Review 38 (2021) 127–137. URL: https://doi.org/10.1177/02663821211034079. doi:10.1177/02663821211034079. arXiv:https://doi.org/10.1177/02663821211034079.

[18] A. MacFarlane, T. Russell-Rose, F. Shokraneh, Search strategy formulation for systematic reviews: Issues, challenges and opportunities, Intelligent Systems with Applications 15 (2022) 200091. URL: https://www.sciencedirect.com/science/article/pii/S266730532200031X. doi:https://doi.org/10.1016/j.iswa.2022.200091.

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

[20] S. Lima-López, E. Farré-Maduell, L. Gascó, A. Nentidis, A. Krithara, G. Katsimpras, G. Paliouras, M. Krallinger, Overview of MedProcNER task on medical procedure detection and entity linking at BioASQ 2023, in: Working Notes of CLEF 2023 - Conference and Labs of the Evaluation Forum, 2023.

[21] M. Q. Stearns, C. Price, K. A. Spackman, A. Y. Wang, SNOMED clinical terms: overview of the development process and project status., in: Proceedings of the AMIA Symposium, American Medical Informatics Association, 2001, p. 662.

[22] S. Wang, H. Li, H. Scells, D. Locke, G. Zuccon, Mesh term suggestion for systematic review literature search, in: Proceedings of the 25th Australasian Document Computing Symposium, ADCS '21, Association for Computing Machinery, New York, NY, USA, 2022. URL: https://doi.org/10.1145/3503516.3503530. doi:10.1145/3503516.3503530.

[23] S. Wang, H. Li, G. Zuccon, Mesh suggester: A library and system for mesh term suggestion for systematic review boolean query construction, in: Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, WSDM '23, Association for Computing Machinery, New York, NY, USA, 2023, p. 1176–1179. URL: https://doi.org/10.1145/3539597.3573025. doi:10.1145/3539597.3573025.

[24] Y. Zhou, A. I. Muresanu, Z. Han, K. Paster, S. Pitis, H. Chan, J. Ba, Large Language Models Are Human-Level Prompt Engineers, 2023. arXiv:2211.01910.

[25] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, P. Fung, Survey

of Hallucination in Natural Language Generation, ACM Comput. Surv. 55 (2023). URL: https://doi.org/10.1145/3571730. doi:10.1145/3571730.

[26] R. Eldan, Y. Li, TinyStories: How Small Can Language Models Be and Still Speak Coherent English?, 2023. arXiv:2305.07759.

[27] N. Ho, L. Schmid, S.-Y. Yun, Large Language Models Are Reasoning Teachers, 2023. arXiv:2212.10071.

[28] U. Kruschwitz, C. Hull, Searching the Enterprise, Foundations and Trends® in Information Retrieval 11 (2017) 1–142. URL: http://dx.doi.org/10.1561/1500000053. doi:10.1561/1500000053.

[29] E. M. Bender, T. Gebru, A. McMillan-Major, S. Shmitchell, On the dangers of stochastic parrots: Can language models be too big?, in: Proceedings of the 2021 ACM conference on fairness, accountability, and transparency, 2021, pp. 610–623.