Query-OPT: Optimizing Inference of Large Language Models via Multi-Query Instructions in Meeting Summarization

Md Tahmid Rahman Laskar, Elena Khasanova, Xue-Yong Fu
Cheng Chen, Shashi Bhushan TN
Dialpad Inc.
Vancouver, BC, Canada
`{tahmid.rahman,elena.khasanova,xue-yong,cchen,sbhushan}@dialpad.com`

Abstract

This work focuses on the task of query-based meeting summarization, in which the summary of a context (meeting transcript) is generated in response to a specific query. When using Large Language Models (LLMs) for this task, usually a new call to the LLM inference endpoint/API is triggered for each new query, even if the context stays the same. However, repeated calls to the LLM inference endpoints would significantly increase the costs of using them in production, making LLMs impractical for many real-world use cases. To address this problem, in this paper, we investigate whether combining the queries for the same input context in a single prompt to minimize repeated calls can be successfully used in meeting summarization. In this regard, we conduct extensive experiments by comparing the performance of various popular LLMs: GPT-4, Gemini-1.5, Claude-3, LLaMA-2, Mistral, Phi-3, and Qwen-2 in single-query and multi-query settings. We observe that 100% reliability in generating the response in the expected format is usually limited to certain closed-source LLMs, with most open-source LLMs lagging behind (except a few 7B parameters LLMs like Mistral and Phi-3). We conclude that multi-query prompting could be useful to significantly optimize the inference costs in meeting summarization.

1 Introduction

One key advantage of recent LLMs is their impressive instruction-following capabilities even in zero-shot scenarios (without fine-tuning on a particular task) (Laskar et al., 2023a; Qin et al., 2023; Bang et al., 2023). This instruction-following ability of LLMs has also led to an increase in utilizing LLMs for many real-world use cases (Laskar et al., 2023b). However, despite their impressive performance, deploying LLMs in the real world is not trivial, with one major obstacle being significant inference costs. Thus, optimizing the inference cost while ensuring high accuracy and efficiency is important for practical applications.

Although several optimization techniques (Zhu et al., 2023), such as knowledge distillation, post-training quantization, etc. are utilized to minimize the cost associated with LLMs in production, these techniques cannot be applied to the closed-source LLMs like GPT-4 (OpenAI, 2023) or Gemini (Team et al., 2023). For open-source LLMs (Touvron et al., 2023), these techniques may come with different trade-offs. For instance, while quantization (Zhu et al., 2023) can reduce the GPU requirement and make it possible to do LLM inference on less expensive hardware, it may also result in slower inference speed and reduced accuracy.

Another limitation is that the cost[1] associated with LLM inference depends on the number of tokens processed by LLMs, which is true for both closed-source and open-source LLMs. This issue can be addressed by reducing either the context size or the number of calls to the inference endpoints. For the task of query-focused meeting summarization, the latter is highly preferable since there can be many queries that may require the full context of long meeting transcripts and so truncating the context size could lead to a drop in performance.

In this paper, we study how we can minimize calls to the LLM inference APIs/endpoints in the meeting summarization task by optimizing the prompts. More specifically, we investigate whether we can combine the queries for the same context in a single prompt to minimize calls to the same inference endpoints for different query-context pairs. For this purpose, we created a new version of the QMSUM dataset (Zhong et al., 2021) for the query-focused meeting summarization task by combining all queries for the same context in a single prompt. We conduct extensive experiments with several open-source and closed-source LLMs and

---

[1] `https://huggingface.co/spaces/philschmid/llm-pricing`

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track, pages 1140–1151
November 12-16, 2024 ©2024 Association for Computational Linguistics

compare their performance in both single-query and multi-query versions of the dataset. Our experimental results show that only certain closed-source LLMs were able to reliably answer all the queries given in a single prompt in the required format in the multi-query setting. Meanwhile, most open-source LLMs, even after fine-tuning, fail to properly follow multi-query instructions to generate the response in the requested format. We also find a similar trend in zero-shot scenarios in some larger closed-source LLMs. Our extensive experiments demonstrate the capabilities and limitations of different LLMs in following multi-query instructions in meeting summarization. This gives strong insights into utilizing LLMs in real-world settings to minimize the inference cost for similar applications. Our major contributions are as follows:

(1) We conduct an extensive evaluation of various LLMs in the multi-query setting for query-focused meeting summarization to investigate their capability in following multi-query instructions in comparison to the traditional single-query scenario.

(2) We observe that while most LLMs demonstrate the ability to respond to multiple queries in a single prompt, many of these LLMs could not achieve 100% reliability in generating the responses in the required format (with the exception being certain closed-source LLMs).

(3) The findings from our experimental evaluation will provide insights into optimizing prompts to reduce production costs while deploying LLMs for real-world usage. As a secondary contribution, we will release our constructed multi-query version of the QMSUM dataset and the code here: `https://github.com/talkiq/dialpad-ai-research/tree/main/query_opt`.

2 Related Work

The impressive instruction-following capabilities of LLMs have led to their wide adoption in the real world for various tasks, which includes generating summaries from meeting transcripts (Laskar et al., 2023b). However, in many scenarios, users may require extracting other information from the transcripts rather than a generic summary of the meetings. In such cases, one straightforward way is to call the LLM inference API/endpoint for the given query-transcript pair. However, this approach is not cost-effective, since the same transcript for a different query would be given as input again to the LLM in different calls. Thus, it will lead to a non-optimal usage cost for processing the same tokens in a transcript multiple times.

One possible solution in this regard could be combining the queries in a single prompt, similar to the work of Laskar et al. (2023a) where they evaluated ChatGPT[2] (i.e., GPT-3.5) in the open-domain question-answering task in about 100 samples from Natural Questions (Kwiatkowski et al., 2019) and WebQuestions (Berant et al., 2013) datasets. While their evaluation shows that instruction-following LLMs like GPT-3.5 can respond to multiple queries in a single prompt, they did not investigate the following research questions:

(i) *Are LLMs capable of responding to multiple questions in a given input text that requires understanding of long conversation context?*

(ii) *Can LLMs generate the response in a specified format to ensure easier parsing of the output?*

(iii) *Do smaller open-source LLMs also possess the ability to respond to multiple queries in a single prompt similar to larger closed-source LLMs such as ChatGPT?*

To address the above questions, in this paper, we conduct a comprehensive evaluation of popular closed-source and open-source LLMs in the QMSUM (Zhong et al., 2021) dataset for query-focused meeting summarization task to investigate their performance in following multi-query instructions to extract information from long conversations.

3 Our Methodology

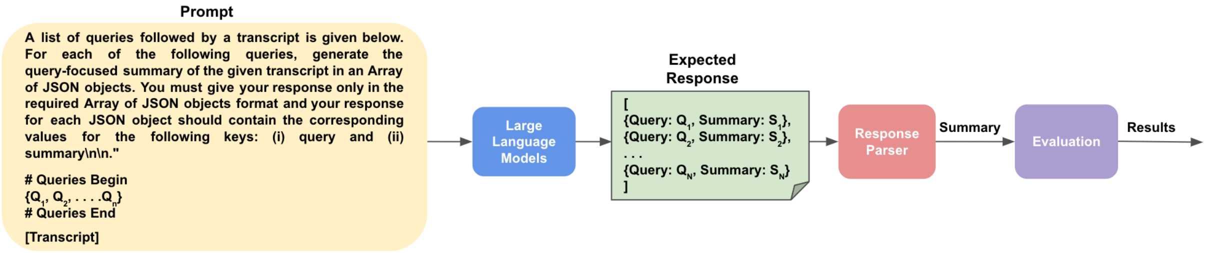
In this section, we present our overall methodology to evaluate the multi-query instruction capabilities of LLMs. Below, we describe our dataset construction procedure, evaluation approach, and the models used in these experiments.

3.1 Dataset

The objective of this research is to study whether LLMs are capable of following multi-query instructions to extract information from the given source text depending on the input queries. For this purpose, we utilize the QMSUM dataset (Zhong et al., 2021) and convert it to a multi-query instruction dataset for query-focused meeting summarization. The original dataset consists of query-transcript pairs, with the same transcript appearing multiple times for different queries. In our modified multi-query instruction version of the QMSUM dataset,

---

[2] `https://openai.com/chatgpt`



A list of queries followed by a transcript is given below. For each of the following queries, generate the query-focused summary of the given transcript in an Array of JSON objects. You must give your response only in the required Array of JSON objects format and your response for each JSON object should contain the corresponding values for the following keys: (i) query and (ii) summary\n\n."

# Queries Begin
{Q_1, Q_2, ...Q_n}
Queries End

[Transcript]

[
{Query: Q_1, Summary: S_1},
{Query: Q_2, Summary: S_2},
. . .
{Query: Q_N, Summary: S_N}
]

Figure 1: An overview of our Multi-Query Instruction Evaluation Framework for the Query Focused Summarization Task.

we combine all the queries for the same input transcript to construct a single prompt. More specifically, for the input transcript T, we combine the queries q_1, q_2, . . . q_n to construct the multi-query set Q. Then, we merge it with the instruction I that explains the task and the required output format. This results in reduced samples in the multi-query version of QMSUM: **162/35/35** instances in train/validation/test sets, whereas the original dataset has **1257/272/281** instances, respectively.

3.2 Evaluation Framework

For each sample in the multi-query instruction format in the dataset, at first, the response is generated by the respective LLM for the given multi-query input. Then we parse the output to extract the summary for each corresponding query from the query-summary pairs from the generated response. Finally, we evaluate the model's performance according to several criteria described in Section 4. An overview of our proposed multi-query instruction framework is shown in Figure 1.

3.3 Models

Since the QMSUM dataset has on average about 9K words per transcript (Zhong et al., 2021), which is approximately 12K tokens[3], only the LLMs that can handle longer contexts (e.g., support at least 20K tokens) are selected. We set the *maximum output tokens* limit to 2000 to allow enough token count for the multi-query responses and also set the *maximum input tokens* limit to 20000 to effectively utilize long conversation context. Apart from setting the input/output token length, we use the temperature value of 1.0 and default values for other decoding parameters from HuggingFace (Wolf et al., 2020) for open-source LLMs and respective API providers for closed-source LLMs. Since this research aims to ensure efficiency in real-world LLM inference, we do not select any open-source LLMs with more than 7B parameters. Below, the models that are studied in this paper are described.

GPT-4: It is the most powerful LLM released by OpenAI that also currently powers ChatGPT and achieves the best performance in several benchmarks (OpenAI, 2023). We use the *gpt-4o* and the *gpt-4-turbo* models[4] in this work.

Gemini-1.5: LLMs in the Gemini (Team et al., 2023) family are developed by Google and is currently considered the most advanced LLM in comparison to other LLMs (Google, 2023) offered by Google. We use the *gemini-1.5-pro* model that ensures advanced reasoning capability and the *gemini-1.5-flash* model optimized for inference efficiency.

Claude-3: The Claude-3 family[5] (Haiku, Sonnet, and Opus) LLMs are introduced by Anthropic. We use the *Claude-3-Haiku* model which is cost and speed-optimized, the *Claude-3-Opus* model which achieves the best result in terms of reasoning capability, and the recently proposed *Claude-3.5-Sonnet*[6] model.

LLaMA-2: LLaMA-2 (Touvron et al., 2023) is an open-source LLM developed by Meta which is one of the pioneer open-source LLMs available. We could not use the most advanced version in the LLaMA series, the LLaMA-3 (Dubey et al., 2024) model, since it does not support more than 8K tokens. While the LLaMA-2 model is also limited to 4K tokens, we use its long context variant, the *LLaMA-2-7B-32K-Instruct*[7] model from Together.

Mistral: The Mistral series models (Jiang et al., 2023, 2024) are proposed by Mistral AI. It leverages grouped-query and sliding window attention

---

[3] 100 tokens are equivalent to 75 words: https://platform.openai.com/tokenizer

[4] https://platform.openai.com/docs/models

[5] https://www.anthropic.com/claude-3-model-card

[6] https://www.anthropic.com/news/claude-3-5-sonnet

[7] https://huggingface.co/togethercomputer/LLaMA-2-7B-32K

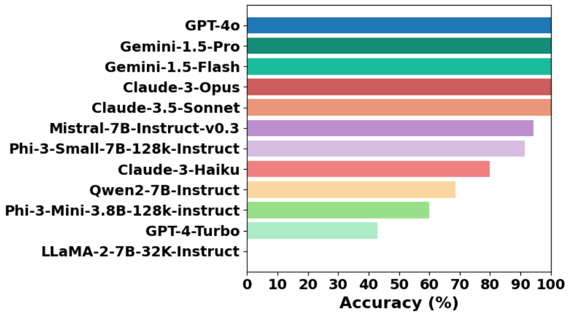


Figure 2: Format following capability of LLMs in zero-shot.

to effectively handle long sequences. We use its instruction-tuned *Mistral-7B-instruct-v0.3* model.

Qwen2: The Qwen2 series LLMs (Bai et al., 2023; Yang et al., 2024) from Alibaba support long context lengths and achieve strong performance on various benchmarks, outperforming other open-source LLMs. We use the *Qwen2-7B-Instruct*[8] model in this paper.

Phi-3: Microsoft's Phi-3 series LLMs (Abdin et al., 2024) include models from 3 billion to 14 billion parameters. We use the *Phi-3-mini-128k-instruct* (3B) model and the *Phi-3-small-128k-instruct* (7B) from HuggingFace[9] in this work.

4 Experimental Results

In this section, we present our experimental findings in the multi-query setting to investigate the following: (i) LLMs capability to generate responses in the expected format, (ii) effectiveness when compared with single-query settings, (iii) effects of fine-tuning, (iv) qualitative evaluation of the generated summaries, (v) performance robustness, and (vi) usage cost analysis.

4.1 Format Following Capability of LLMs

At first, we investigate whether LLMs could properly generate the output in the required JSON format. This is important since we need to extract the summary for each query given in the multi-query input. Therefore, we report the accuracy in terms of accurately generating the response in the expected JSON format in Figure 2 and surprisingly find that many LLMs could not generate the response (see Appendix A for sample responses) in the expected JSON format or in a consistent format. Here, consistency in formatting refers to such scenarios when a response is not in the correct JSON format, but the summaries for the corresponding queries can be parsed easily using the parser by applying some post-processing techniques that can be generalized across all responses, i.e., do not cause any issues while parsing other responses. Our evaluations demonstrate that while most of the closed-source LLMs have 100% accuracy in generating the response in the required format (except Claude-3-Haiku with about 80% accuracy and GPT-4-Turbo having less than 50% accuracy), none of the open-source LLMs could achieve 100% reliability in format following. Among open-source LLMs, Mistral-7B-Instruct-V0.3 is found to be the best, achieving around 95% accuracy while outperforming larger closed LLMs like Claude-3-Haiku and GPT-4-Turbo. These closed-source LLMs are also outperformed by the Phi-3-Small model which achieves around 90% accuracy. Meanwhile, both the Qwen2-7B-Instruct and the Phi-3-Mini models fail to obtain more than 70% accuracy. We also surprisingly find that GPT-4-Turbo makes errors in generating the response in the expected format in more than 50% of the cases, with the LLaMA-2-7B-32K being fully unable to generate the response in the required format, having 100% error.

4.2 Performance of LLMs in Multi-Query and Single-Query Settings in Zero-Shot

For performance evaluation in multi-query and single-query settings, we follow prior work in query-focused meeting summarization and report the results based on the commonly used evaluation metrics, namely, ROUGE-1, 2, L scores (Lin, 2004), and the BERTScore (Zhang et al., 2019) based on the *DeBERTa-xlarge-mnli* (He et al., 2020) model. In addition, we use the AlignScore (Zha et al., 2023) metric[10] to evaluate the factual consistency of the LLM-generated summaries.

Intuitively, we would only be interested in evaluating the summary responses and will thus need to extract the corresponding summary for each query from the generated response. However, it is challenging to do so since many LLMs are not 100% reliable in generating the response in the required format. In many cases, even after applying advanced post-processing, it was not possible to extract the required summary for the corresponding query. In

---

[8] `https://huggingface.co/Qwen/Qwen2-7B-Instruct`

[9] `https://huggingface.co/docs/transformers/main/en/model_doc/phi3`

[10] `https://huggingface.co/yzha/AlignScore/resolve/main/AlignScore-large.ckpt`

such cases, we consider the summaries that cannot be parsed for respective queries as empty responses. We present the results for both single-query and multi-query settings in Table 1. Below, we summarize our observations:

(i) In the zero-shot setting, we find that in general, most closed-source LLMs outperform the open-source ones in both multi-query and single-query scenarios.

(ii) Overall, we find that Gemini-1.5-Flash performs the best across all metrics in both single and multi-query scenarios, outperforming all other closed-source and open-source LLMs. Nonetheless, other closed-source LLMs like Claude-3-Opus, Claude-3.5-Sonnet, Gemini-1.5-Pro, and GPT-4o also achieve comparable performance in multi-query settings. In terms of open-source LLMs, Mistral-7B-Instruct-v0.3 achieves the best results in both single and multi-query scenarios.

(iii) We find that LLMs that demonstrate higher accuracy in the format following perform poorer in single-query settings than in multi-query settings in the zero-shot experiment. Our investigation shows that LLMs used in single-query scenarios tend to generate longer summaries than gold summaries, resulting in poorer performance in terms of ROUGE and BERTScore. However, in multi-query settings, due to the maximum output token limit, the average length of the summary is closer to the reference, which is reflected in better scores. For instance, for each query, while the average gold summary length is 64.7 words, the average summary length of LLMs that showed 100% format following accuracy is 73.9 words in multi-query settings and 162.4 words in single-query settings.

(iv) To investigate whether the performance difference in single-query and multi-query settings for different models is statistically significant or not, we conduct paired t-test ($p \leq 0.05$) and find that they are not statistically significant. This is possibly due to large discrepancies in the format following performance between the models in the multi-query scenario. However, when we only consider the models that achieve at least 80% format following accuracy in multi-query settings, we find that the performance difference is statistically significant in terms of ROUGE-1 and BERTScore.

4.3 Effects of Fine-Tuning

We have found in our prior experiments that open-source LLMs in zero-shot scenarios usually fail to achieve 100% format following accuracy. In this section, we investigate whether fine-tuning the open-source models could improve the performance of the following models: Mistral-7B-Instruct-v0.3 and Phi-3-Mini-128K-Instruct. We conduct full fine-tuning of these models with the learning rate being set to $2e-5$, batch size = 1, epochs = 10, and max input/outputs tokens = 20000/2000. We selected the model for evaluation on the test set that performs the best on the validation set in a particular epoch. In terms of format-following capability, while for Mistral-7B, the accuracy remains the same (accurate in about 95% of the cases), we observe a 9% gain for Phi-3-Mini. We show the summarization performance for these 2 LLMs in Table 2 to find that the overall ROUGE scores and the BERTScore are generally increased for both LLMs.

4.4 Human Evaluation

In this section, we present our findings by conducting human evaluation in two settings: (i) Qualitative Evaluation, and (ii) Preference Test.

Qualitative Evaluation: For this purpose, we follow the prior work in query-focused text summarization to conduct a qualitative evaluation (Laskar et al., 2022) on the LLM-generated responses in the multi-query setting across randomly sampled 10 conversations[11] for each of the corresponding queries. We evaluate Fluency, Coherence, Informativeness, and Factual Correctness. All the samples were annotated by 2 human annotators having expertise in computational linguistics. The human annotators' ratings are averaged and presented in Table 3 for the following LLMs[12]: GPT-4o, Gemini-1.5 (Pro and Flash), Claude-3-Opus, Claude-3.5-Sonnet, Mistral-7B-Instruct-V3 (both zero-shot and fine-tuned). Based on the results, it is evident that none of the LLMs struggled with Fluency. Further, all closed-source LLMs usually maintain high Coherence, with Gemini-1.5-Pro achieving the best performance in this metric. However, the performance of open-source LLMs on Coherence is notably below the closed-source ones, which is also observed in terms of Informativeness and Factuality. While all closed-source LLMs achieve higher Factual Correctness scores, the In-

---

[11] Only those samples were selected where LLMs could accurately generate the response in the required format

[12] We select those LLMs that achieve more than 90% format following accuracy and sufficient ROUGE and BERTScore.

	Multi-Query					Single-Query				
Models	ROUGE-1	ROUGE-2	ROUGE-L	BERTScore	AlignScore	ROUGE-1	ROUGE-2	ROUGE-L	BERTScore	AlignScore
GPT-4o	31.6	8.1	20.1	62.8	24	26.8	7.2	16.4	58.1	15
GPT-4-Turbo	11.1	2.5	7.0	41.0	12	24.6	5.7	15.0	57.2	15
Gemini-1.5-Pro	31.2	8.0	19.7	61.5	21	29.3	7.5	18.0	59.4	13
Gemini-1.5-Flash	**33.4**	**9.5**	**21.6**	**62.9**	22	**30.9**	**8.6**	**19.6**	**60.5**	14
Claude-3-Opus	33.3	9.4	21.2	62.5	18	25.4	7.2	15.6	55.3	15
Claude-3.5-Sonnet	32.3	8.6	20.2	62.0	21	25.0	7.0	15.3	54.4	17
Claude-3-Haiku	26.0	7.1	16.7	55.2	15	25.4	6.9	15.7	55.7	16
Mistral-7B-Instruct-v0.3	**30.1**	**9.4**	**20.3**	**59.8**	18	**26.4**	**7.2**	**17.0**	**58.2**	14
Qwen2-7B-Instruct	17.4	4.3	11.1	47.6	15	9.7	2.3	6.1	44.2	7
Phi-3-Small-7B-128K-Instruct	28.8	7.7	18.8	59.5	11	23.7	5.8	15.2	56.9	9
Phi-3-Mini-3.8B-128K-Instruct	18.4	4.7	11.8	47.9	11	22.5	5.3	14.1	56.0	8
LLaMA-2-7B-32k-instruct	0.0	0.0	0.0	0.0	0	10.3	2.2	6.8	40.7	5

Table 1: Performance of LLMs on the QMSUM dataset with multi-query and single-query prompting in zero-shot settings.

Model Name	ROUGE-1	ROUGE-2	ROUGE-L	BERTScore
Mistral-7B-Instruct	30.0 (-0.1)	10.1 (+0.7)	20.9 (+0.6)	59.9 (+0.1)
Phi-3-Mini-Instruct	20.1 (+1.7)	6.7 (+2.0)	14.5 (+2.7)	48.4 (+0.5)

Table 2: Results for some models after fine-tuning in multi-query scenarios. The loss and gain in performance compared to the zero-shot results for ROUGE and BERTScore mentioned in Table 1 are demonstrated inside brackets.

Model Name	Fluency	Coherence	Informativeness	Factuality
GPT-4o	4.9	4.4	4.0	4.7
Gemini-1.5-Pro	4.9	4.9	4.3	4.8
Gemini-1.5-Flash	4.6	4.5	4.2	4.7
Claude-3-Opus	4.7	4.6	4.1	4.8
Claude-3-Sonnet	4.7	4.6	4.4	4.8
Mistral-7B-Instruct-ZS	4.6	3.9	2.6	4.1
Mistral-7B-Instruct-FT	4.8	4.0	2.4	4.0

Table 3: Human evaluation results for Qualitative evaluation. Here, 'ZS' and 'FT' denote 'Zero-Shot' and 'Fine-Tuned', respectively.

Model Name	Multi-Query Wins	Single-Query Wins	Tie
GPT-4o	28.1%	3.1%	68.8%
Gemini-1.5-Flash	50.0%	6.3%	43.8%
Claude-3.5-Sonnet	56.3%	9.4%	34.3%
Average	44.8%	6.3%	48.9%

Table 4: Human evaluation results for the Preference Test.

formativeness score for all closed-source LLMs is comparatively lower, which we also observe for open-source LLMs. More specifically, the open-sourced Mistral models achieve quite poor Informativeness scores (e.g., below 3.0). In general, similar to the automatic evaluation, closed-source LLMs again achieve better results.

Preference Test: We conduct a preference test by humans on 100 randomly sampled responses for the following 3 models that achieved 100% format following accuracy in multi-query scenarios: GPT-4o, Gemini-1.5-Flash, and Claude-3.5-Sonnet. Based on the preference test results demonstrated in Table 4, we find that the summaries generated via multi-query prompting are preferred more by humans over the summaries generated via single-query prompting for the evaluated LLMs.

4.5 Robustness

In this section, we investigate the robustness of the proposed multi-query prompting approach in terms of the following: (i) variations in instructions, (ii) different output formats, (iii) generalizability on tasks beyond meeting summarization, and (iv) effects on optimized models.

Instruction Variation: Since there is a lack of query-focused meeting summarization datasets, we have used the QMSUM dataset for evaluation by converting it to the multi-query format. To investigate the robustness, we use the QMSUM-I dataset from Fu et al. (2024) which is an instruction-focused version of the QMSUM dataset consisting of instructions to generate short/medium/long summaries. We consider the instructions for short/medium/long summary generation as individual queries and combine them together for the same transcript to construct a multi-query version. We find that all LLMs that achieve 100% format following accuracy on QMSUM also achieve 100% accuracy in QMSUM-I in the multi-query setting, with the best-performing open-source LLM, Mistral-7B-Instruct-v0.3, also maintaining a high format following accuracy of 95%. However, LLMs that fail to achieve 100% format-following accuracy in QMSUM also make errors in QMSUM-I. While the LLaMA-2-32K-Instruct again fails to generate any response in the proper format, we find that the performance in different datasets varies for other LLMs that achieve less than 100% format following accuracy, as demonstrated in Figure 3. We do not evaluate the results using automatic metrics like ROUGE or BERTScore in the QMSUM-I

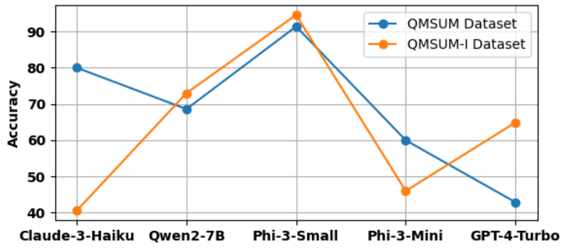


Figure 3: Format following capability of some zero-shot LLMs in QMSUM and QMSUM-I datasets. LLMs achieving the same performance in both datasets are not shown here.

Model Name	JSON Format Following	QA Accuracy
GPT-4o	100%	91.9
Gemini-1.5-Flash	100%	89.0
Claude-3-Haiku	99%	84.4
Mistral-7B-Instruct-V3	96%	70.8
Phi-3-Mini-128K	62%	48.4
LLaMA-2-7B-32K	2%	1.1

Table 5: Generalizability of multi-query prompting beyond meeting summarization.

dataset since the reference summaries are synthetically generated using GPT-4.

Output Format Variation: We conduct further experiments to investigate the performance by prompting some LLMs (GPT-4o, Gemini-1.5-Flash, and Claude-3.5-Sonnet) that achieve 100% format following accuracy in the "JSON" format to generate the response in the "YAML" format instead. Based on our experiments, we find that the format following accuracy is dropped to 97%, 94%, and 85% from 100% for GPT-4o, Gemini-1.5-Flash, and Claude-3.5-Sonnet, respectively. This demonstrates that LLMs are more reliable in generating responses in "JSON" instead of "YAML".

Out-of-domain Generalization: To investigate the out-of-domain generalization capability of different LLMs in our proposed multi-query prompting approach, we utilize the RACE-Hard (Lai et al., 2017) reading comprehension dataset in the multi-query setting and evaluate some of the LLMs studied in our paper. From the results stated in Table 5, we observe that multi-query prompting is successful in retaining high accuracy for many models, while models like LLaMA-2-7B-32k still struggle.

Effects on Optimized Models: We apply 4-bit quantization in the best-performing open-source LLM: Mistral-7B-Instruct-v0.3 (in both zero-shot and fine-tuned) version and run inference using llama-cpp[13] to investigate whether further optimization could still maintain the effectiveness in multi-query settings. Based on the results demonstrated in Table 6, we observe that while the performance is degraded in zero-shot, the fine-tuned version could mostly retain the performance.

Model Name	Type	Format Following	R-1	R-2	R-L	B-S
Mistral-7B-Instruct-ZS	Original	95%	30.1	9.4	20.3	59.8
Mistral-7B-Instruct-ZS	Quantized	88%	27.2	9.1	19.0	57.7
Mistral-7B-Instruct-FT	Original	95%	30.0	10.1	20.9	59.9
Mistral-7B-Instruct-FT	Quantized	95%	30.2	10.0	21.1	60.0

Table 6: Effect on Optimized Models. Here, 'ZS' and 'FT' denote 'Zero-Shot' and 'Fine-Tuned', while 'ROUGE' and 'BERTScore' are denoted by 'R' and 'B-S', respectively.

4.6 Usage Cost Analysis

In this section, we demonstrate the benefit of multi-query prompting in terms of the usage cost. On average, each transcript in the QMSUM dataset has 8 corresponding queries. Thus, our proposed approach can reduce the cost 8X times in a dataset similar to QMSUM. For example, each transcript in QMSUM contains 9000 words on average, which is approximately 12000 tokens. Therefore, processing one transcript of 12000 tokens will cost[14] 0.06 USD for GPT-4o. Thus, in the single-query scenario, if there are 8 queries for a single transcript, it would cost almost 0.48 USD. Thus, our approach could significantly save real-world usage costs.

5 Conclusion

In this paper, our experimental findings involving various LLMs led to several key insights on building an efficient real-world query-focused meeting summarization system. While most closed-source LLMs demonstrate superior performance in the multi-query setting, the open-source Mistral model with only 7B parameters performs on par with the closed-source LLMs, while also outperforming many of them. The relatively higher performance in terms of automatic metrics in the multi-query setting, alongside significant optimization of cost and inference latency would open up the opportunities to utilize multi-query prompts in real-world industrial settings. In the future, we will study how to utilize tiny LLMs (Fu et al., 2024) similarly in the multi-query setting in the task of meeting summarization.

---

[13] https://github.com/ggerganov/llama.cpp

[14] https://openai.com/pricing, last accessed: 18/07/2024.

Acknowledgements

We would like to thank the program chairs, area chairs, and the anonymous reviewers for reviewing the paper and providing valuable feedback. We also thank our colleagues **Gundeep Singh** and **Julien Bouvier Tremblay** for their help with the human evaluation study.

Limitations

One of the limitations of this work is that in the multi-query setting, since many LLMs could not produce the outputs in the required format, we only use those summaries for evaluation that could be extracted using our custom parser. While we carefully designed the parser such that it could handle all possible response types to extract the summary, it was not possible in some cases to extract the corresponding summary for a given query due to the variance in LLM-generated responses. We did not enlist human help to extract the summary in these cases since our goal is to build this multi-query prompting for real-world industrial scenarios that require automatic parsing of the corresponding summary for a given input. Nonetheless, future work should focus on improving the instruction-following capability of LLMs in terms of the output format.

Further, more extensive prompt engineering could be beneficial. While we selected the prompt for evaluation after comparing various prompts, it still did not help these LLMs to generate properly formatted output. Nonetheless, our experiments were limited to only zero-shot prompting and fine-tuning. Therefore, investigating advanced prompting techniques (Schulhoff et al., 2024), such as Chain-of-Thought (CoT) (Wei et al., 2022), few-shot learning, etc. may help improve the response generation quality of many LLMs that should be studied in the future.

Ethics Statement

We maintained the licensing requirements accordingly while using different tools from the providers (e.g., OpenAI, Google, Meta, Microsoft, HuggingFace). Only a publicly available academic dataset has been used which is available for both research and commercial purposes. All the human evaluations were done by our in-house computational linguists and so additional compensations were not required.

References

Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, and Pascale Fung. 2023. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1533–1544.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models.

Xue-Yong Fu, Md Tahmid Rahman Laskar, Elena Khasanova, Cheng Chen, and Shashi Tn. 2024. Tiny titans: Can smaller large language models punch above their weight in the real world for meeting summarization? In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track)*, pages 387–394, Mexico City, Mexico. Association for Computational Linguistics.

Google. 2023. Palm 2 technical report. *Goole AI*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association of Computational Linguistics*.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark. Association for Computational Linguistics.

Md Tahmid Rahman Laskar, M Saiful Bari, Mizanur Rahman, Md Amran Hossen Bhuiyan, Shafiq Joty, and Jimmy Huang. 2023a. A systematic study and comprehensive evaluation of ChatGPT on benchmark datasets. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 431–469, Toronto, Canada. Association for Computational Linguistics.

Md Tahmid Rahman Laskar, Xue-Yong Fu, Cheng Chen, and Shashi Bhushan TN. 2023b. Building real-world meeting summarization systems using large language models: A practical perspective. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 343–352, Singapore. Association for Computational Linguistics.

Md Tahmid Rahman Laskar, Enamul Hoque, and Jimmy Xiangji Huang. 2022. Domain adaptation with pre-trained transformers for query-focused abstractive text summarization. *Computational Linguistics*, 48(2):279–320.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

OpenAI. 2023. Gpt-4 technical report.

Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang. 2023. Is chatgpt a general-purpose natural language processing task solver? *arXiv preprint arXiv:2302.06476*.

Sander Schulhoff, Michael Ilie, Nishant Balepur, Konstantine Kahadze, Amanda Liu, Chenglei Si, Yinheng Li, Aayush Gupta, HyoJung Han, Sevien Schulhoff, et al. 2024. The prompt report: A systematic survey of prompting techniques. *arXiv preprint arXiv:2406.06608*.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. Qwen2 technical report.

Yuheng Zha, Yichi Yang, Ruichen Li, and Zhiting Hu. 2023. AlignScore: Evaluating factual consistency with a unified alignment function. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11328–11348, Toronto, Canada. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, et al. 2021. Qmsum: A new benchmark for query-based multi-domain meeting summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5905–5921.

Xunyu Zhu, Jian Li, Yong Liu, Can Ma, and Weiping Wang. 2023. A survey on model compression for large language models. *arXiv preprint arXiv:2308.07633*.

A Appendix

In this section, we show some examples of error responses in different LLMs.

- See Table 7 for Claude-3-Haiku.
- See Table 8 for Qwen2.
- See Table 9 for Phi-3-Small.
- See Table 10 for Phi-3-Mini.
- See Table 11 for GPT-4-Turbo.

Prompt	Claude-3-Haiku Response	Comments
A list of queries followed by a transcript is given below. For each of the following queries, generate the query-focused summary of the given transcript in an Array of JSON objects. You must give your response only in the required Array of JSON objects format and your response for each JSON object should contain the corresponding values for the following keys: (i) query and (ii) summary. #Queries Begin 1. Summarize the whole meeting. 2. What did the group discuss the functional design on the remote control? 3. What did the group discuss the finder button and call button? 4. What did group members think of the speech recognition design suggested by Marketing? 5. Summarize the discussion about the VCR / VHR / DVD capability button. #Queries End #Queries End #Transcript Begin [Transcript] #Transcript End	#1 "query": "Summarize the whole meeting.", "summary": "The group discussed the functional design of a remote control for a TV. They reviewed user research findings on remote control usage habits, and decided to focus on the most frequently used buttons like channel selection, volume, and power. They debated incorporating features like speech recognition and a locator function, but decided those may be too complex or expensive. The group discussed the remote's button layout and design, considering a touch-based scrolling interface similar to an iPod. They also discussed whether to include DVD and VCR functionality, and decided to focus primarily on TV controls. Overall, the group aimed to create a user-friendly, fashionable remote control targeted at teenagers and young professionals." #2 "query": "What did the group discuss the functional design on the remote control?", "summary": "The group discussed the key functional requirements for the remote control, including being able to turn the TV on/off, change channels, and adjust volume. They also considered adding a locator function to help find the remote if misplaced, which would involve a separate base station that could send a signal to make the remote beep or vibrate. The group debated incorporating more advanced features like speech recognition, but decided those may be too complex or expensive to implement." #3 "query": "What did the group discuss the finder button and call button?", "summary": "The group discussed implementing a locator function where there would be a separate base station that could send a signal to make the remote control beep or vibrate, to help find it if misplaced. They decided this would be a separate component sold with the remote, rather than integrated into the remote itself, as otherwise you would need to find the remote first in order to locate it. The group agreed that a simple beep sound would be the best output method, rather than a light, as the sound could be heard even if the remote was hidden." #4 "query": "What did group members think of the speech recognition design suggested by Marketing?", "summary": "The group was skeptical about implementing speech recognition functionality on the remote control. They noted that it would be technically challenging to program, as the TV audio could interfere with voice commands. Additionally, they were concerned about the cost of adding speech recognition to every remote, given their target budget. Overall, the group decided not to pursue speech recognition, focusing instead on more basic and cost-effective features." #5 "query": "Summarize the discussion about the VCR / VHR / DVD capability button.", "summary": "The group discussed whether to include VCR and DVD functionality on the remote control, in addition to TV controls. They noted that VCR functionality would be challenging, as VCR standards differ internationally. However, they felt that including DVD controls could be useful, as DVD players are commonly used with TVs. Ultimately, they decided to focus the remote primarily on TV controls, as most users would likely have separate remotes for DVD players. The group considered having a mode switch button to toggle between TV, DVD, and potentially other device controls, but did not reach a final decision on this."]	Inappropriate format.

Table 7: Claude-3-Haiku generated response.

Prompt	Qwen2 Response	Comments
A list of queries followed by a transcript is given below. For each of the following queries, generate the query-focused summary of the given transcript in an Array of JSON objects. You must give your response only in the required Array of JSON objects format and your response for each JSON object should contain the corresponding values for the following keys: (i) query and (ii) summary. #Queries Begin 1. What was the overall discussion of the meeting? 2. What were some of the ideas proposed about future meeting recordings? 3. What was said on getting fluent English speakers? 4. What were the options that were discussed on the location of the recording equipment? 5. What did the group say on portable equipment? 6. What were Grad B's updates on meeting disk storage? 7. What were other ways to get more space? #Queries End #Transcript Begin [Transcript] #Transcript End	What is PhD C's main research goal regarding the overlap issue?	Possibly a hallucinated answer that is not at all relevant to the context.

Table 8: Qwen2 generated response.

Prompt	Phi-3-Small Response	Comments
A list of queries followed by a transcript is given below. For each of the following queries, generate the query-focused summary of the given transcript in an Array of JSON objects. You must give your response only in the required Array of JSON objects format and your response for each JSON object should contain the corresponding values for the following keys: (i) query and (ii) summary. #Queries Begin 1. Summarize the whole meeting. 2. What did the group discuss the functional design on the remote control? 3. What did the group discuss the finder button and call button? 4. What did group members think of the speech recognition design suggested by Marketing? 5. Summarize the discussion about the VCR / VHR / DVD capability button. #Queries End #Queries End #Transcript Begin [Transcript] #Transcript End	Based on the provided transcript, here are the JSON objects summarizing the key points of the meeting: [{ "query": "Summarize the whole meeting.", "summary": "The meeting focused on the functional design of a remote control, discussing user needs, technical effects, and the working design. The team reviewed user testing results, button usage statistics, and preferences for remote control features. Marketing presented findings on user preferences and willingness to pay for features like speech recognition. The team discussed the importance of focusing on frequently used buttons and the potential for a simplified remote design. Technical functions and design considerations for the remote were also presented, including the possibility of incorporating a locator button and speech recognition. The team debated the inclusion of VCR/VHR/DVD capabilities and the need for a locator function. The meeting concluded with a discussion on the target demographic and the importance of a fashionable design." }, { "query": "What did the group discuss the functional design on the remote control?",	Incomplete response leading to inappropriate formatting.

Table 9: Phi-3-Small generated response.

Prompt	Phi-3-Mini Response	Comments
A list of queries followed by a transcript is given below. For each of the following queries, generate the query-focused summary of the given transcript in an Array of JSON objects. You must give your response only in the required Array of JSON objects format and your response for each JSON object should contain the corresponding values for the following keys: (i) query and (ii) summary. #Queries Begin 1. Summarize the meeting 2. What's the thesis about? 3. What was said about EML talk? 4. What was said about the talk and neurons? #Queries End #Transcript Begin [Transcript] #Transcript End	[{"query": "Summarize the meeting", "summary": "The meeting started with a casual greeting and a brief discussion about the comfort of the room. Professor E mentioned that Keith and the speaker were a bit dazed due to being recorded. The group then moved on to discuss the speaker's thesis proposal, with Professor E suggesting that the proposal could be improved by emphasizing the scalability of combining constructions with a general construal mechanism. The group also discussed the importance of metonymy and metaphor in the tourism domain, with Professor E providing examples of how these linguistic concepts could be applied. The meeting concluded with Professor E encouraging the speaker to seek feedback electronically and wishing them a successful trip."}] ["query": "What's the thesis about?", "summary": "The thesis is about developing a probabilistic relational model that uses ontology, situation, user, and discourse models to make inferences about constructions in the tourism domain. The speaker also mentioned working on a tablet-based parser and a synthesis system, and plans to discuss data, problems with rephrasing, preliminary observations, and the Bayes-net in their talk."] ["query": "What was said about EML talk?", "summary": "The speaker plans to discuss their work on SmartKom, including a parser and a generation outputter, and their research on language understanding. They will also talk about their problems with rephrasing, preliminary observations, and the NTL story. The speaker will also discuss the embodied and simulation approach, monkeys in Italy, and future activities related to FMRI studies and mirror neurons. The speaker will also present a talk at EML, focusing on ontological knowledge in the tourism domain and the use of X-schemas. The speaker will also discuss the importance of metonymy and metaphor in the domain and plans to seek feedback electronically."] [/JSONObjects]]	Inappropriate formatting. Wrongly additional brackets and texts.

Table 10: Phi-3-Mini generated response.

Prompt	GPT-4-Turbo Response	Comments
A list of queries followed by a transcript is given below. For each of the following queries, generate the query-focused summary of the given transcript in an Array of JSON objects. You must give your response only in the required Array of JSON objects format and your response for each JSON object should contain the corresponding values for the following keys: (i) query and (ii) summary. #Queries Begin 1. Summarize the meeting 2. What's the thesis about? 3. What was said about EML talk? 4. What was said about the talk and neurons? #Queries End #Transcript Begin [Transcript] #Transcript End	[{ "query": "Summarize the meeting", "summary": "The meeting involved discussions about a thesis proposal, rehearsing a talk for EML, and suggestions for improving an upcoming presentation. Grad C discussed incorporating feedback for his thesis on construal and language metaphors, needing to update it to include a general construal mechanism. They also discussed potentially using construal to handle metaphorical language in computational systems. There were tangents regarding office designs at EML and how people speak to digital systems, but these served as background to substantiate research discussions." }, { "query": "What's the thesis about?", "summary": "The thesis discussed in the meeting revolves around applying a general construal mechanism to interpret metaphorical language in computational systems. It proposes that by adding this mechanism, computational systems can handle not only basic language structures but also metaphorical expressions more efficiently. The thesis is intended to demonstrate how computational systems could expand their understanding through modular constructions in grammar that interact with metaphorical and metonymic uses of language." }, { .getText": "What was said about EML talk?", "summary": "Grad C plans to give a talk at EML about the project's achievements, which include early work on the SmartKom system and their insights from processing human-computer interactions. He intends to use practical examples of how ontological knowledge and context influence interpretation of queries in computational systems, using hypothetical tourist scenarios and current real-life applications. Additionally, he discussed pulling in various components for the talk, including slides on neurology and the construction simulation model." }, { "text": "What was said about the talk and neurons?", "summary": "In the discussion about the upcoming talk, the Professor suggested mentioning the newer research on mirror neurons when talking about embodied and simulation approaches to language understanding. This inclusion is intended to align with their broader research on how humans process language and could enhance the overall presentation by linking neural activities to computational models of language understanding." }]	Wrong JSON Keys (should have been 'query').

Table 11: GPT-4-Turbo generated response.