How Accurate Are LLMs at Multi-Question Answering on Conversational Transcripts?

Xiliang Zhu, Shi Zong, David Rossouw
Dialpad Inc.
{xzhu,shi.zong,davidr}@dialpad.com

Abstract

Deploying Large Language Models (LLMs) for question answering (QA) over lengthy contexts is a significant challenge. In industrial settings, this process is often hindered by high computational costs and latency, especially when multiple questions must be answered based on the same context. In this work, we explore the capabilities of LLMs to answer multiple questions based on the same conversational context. We conduct extensive experiments and benchmark a range of both proprietary and public models on this challenging task. Our findings highlight that while strong proprietary LLMs like GPT-4o achieve the best overall performance, fine-tuned public LLMs with up to 8 billion parameters can surpass GPT-4o in accuracy, which demonstrates their potential for transparent and cost-effective deployment in real-world applications.

1 Introduction

Question Answering (QA) is a pivotal task in Natural Language Processing (NLP), which involves extraction or generation of responses from a given context in reply to user queries. It has been proven invaluable in a diverse array of applications across various industries, such as healthcare, finance, legal and customer support (Jin et al., 2020; Li et al., 2024; Chen et al., 2024; Castelli et al., 2020).

One real-world use case of QA in the customer support domain is producing responses to a set of questions about the agent-customer conversations. This allows for the identification of key components and topics within a conversation in the customer support scenario, helping locate the relevant discussion in the conversation efficiently. Such QA systems enable contact center supervisors to pinpoint potential coaching opportunities, improving agents' ability to better respond to customers' queries and navigate the conversation flow. For instance, verification of the customer's information is often a standard protocol in contact center calls. A question like "*Was the customer's account information verified?*" helps contact center supervisors identify if such protocol is followed in the calls, thus necessary coaching and training can be provided. This approach not only improves operational efficiency but also improves the overall quality of customer service.

With the rapid development of Large Language Models (LLMs), there has been an increasing interest in utilizing LLMs in the QA task. However, deploying LLMs in an industrial production environment presents significant challenges, primarily due to high inference costs and latency. In the context of our QA use case within the customer support domain, we often have multiple questions associated with a single, often lengthy, conversation transcript (over 25% of transcripts contain more than 2,700 tokens; more details in §4.1). Running separate inferences for each question with the same long contextual background creates considerable overhead for the QA system. Therefore, optimizing the inference workflows without compromising the accuracy is crucial for real-world applications.

One natural and intuitive idea to address the above challenge is to combine multiple questions within one single run. Some prior work indeed takes this approach, for example, batch prompting introduced by Cheng et al. (2023); Lin et al. (2024). However, several research questions still remain open. First, to the best of our knowledge, existing empirical evaluations have not considered realistic QA scenarios involving lengthy, multi-turn conversational contexts such as customer-support call-center transcripts. It thus remains unclear whether batch prompting effectively scales to industrial settings, where a single transcript often spans thousands of tokens and triggers dozens of questions. Second, existing studies on batch prompting focus primarily on proprietary LLM APIs with limited

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track, pages 1848–1855

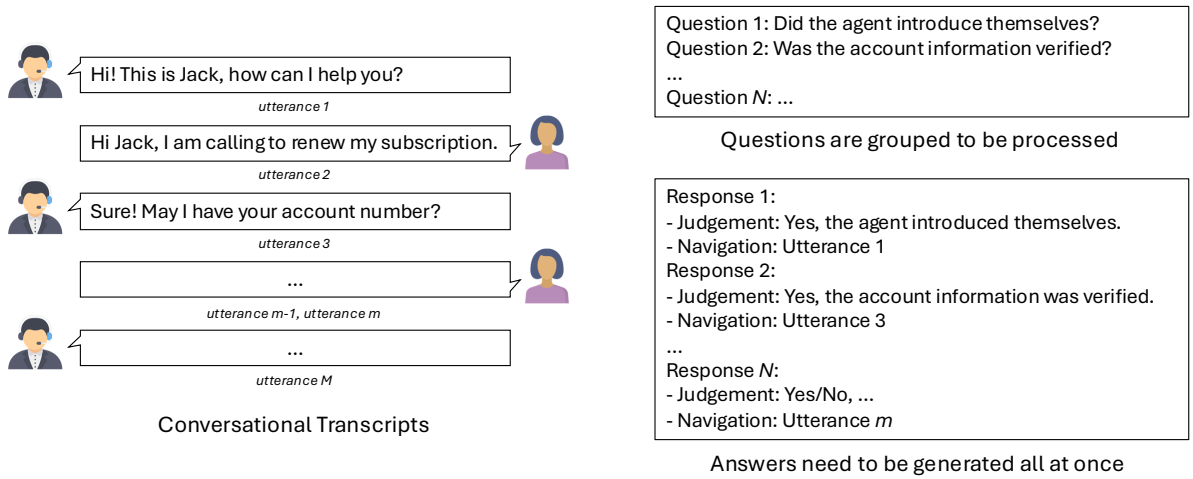

Figure 1: Illustration of our task. The inputs of the model are conversational transcripts in contact centers and questions of group size N, and then we instruct the model to generate the responses to these questions in a single prompt. The model is instructed to generate each response by providing a "yes" or "no" judgment (Judgment) and the supporting utterance from the original transcripts (Navigation).

in-context learning, leaving the potential of public and smaller models unexplored.

In this work, we focus on understanding the capability of LLMs for generating multiple structured responses to a set of questions within a single prompt, based on real-world conversational transcripts. Each response not only includes the answer to the question (Judgment), but also a verifiable reference from the conversation context to enable human review and mitigate hallucination (Navigation). We conduct extensive experiments with several proprietary and public LLMs, and benchmark their performance under both zero-shot and fine-tuned settings.

In summary, our contributions are as follows:

- We adapt the recently-proposed batch prompting paradigm and explore its effectiveness on real-world long conversational transcripts.
- We conduct a comprehensive benchmark comparing leading proprietary LLMs (e.g. GPT-4o) against fine-tuned public LLMs (e.g. Llama-3) on this conversation-based QA task, assessing their ability to process up to 50 questions within a single prompt.
- Our extensive experiments demonstrate that fine-tuned public models with as few as 8 billion parameters can surpass the Judgment accuracy of GPT-4o, which reveals the remarkable potential of smaller public models.

We hope this study will be a useful guide for industry practitioners in making informed decisions regarding model selection and batch size configuration when deploying large-scale QA pipelines for extended customer-support dialogues.

2 Related Work

There has been a long history of studying the QA task. Based on the information source where answers are from, the QA task could be categorized into two types: textual QA and knowledge base QA (Zhu et al., 2021).

Over the years, some other variations have been proposed, including multi-hop QA (Yang et al., 2018), visual QA (Antol et al., 2015), and table QA (Mueller et al., 2019). In this section, we focus on making distinctions between our task and existing tasks in literature and refer readers to recent surveys (for example Zhu et al. (2021); Lan et al. (2023)) for a more comprehensive review of different QA tasks.

Our task is based on conversational transcripts collected in contact centers. It is different from conversational QA in literature (Reddy et al., 2019; Zaib et al., 2022), which engages users in multiple rounds of questions and answers with the model. Moreover, these transcripts are often lengthy and noisy as speech recognition errors are often present. Due to the conversational nature of the transcripts, these contents are also less structured. All of these bring unique challenges compared to the long-context QA that normally uses documents as main resources (Wang et al., 2024).

In our product use case, we primarily focus on

questions that can be directly answered by "yes" or "no". Although this type of question has been previously studied by Yes/No QA (Clark et al., 2019), our main focus is on generating responses for all questions *within one single run* under an industrial environment, instead of processing questions one by one. Additionally, to provide further explainability of the generated binary judgment, we also instruct models to select supporting evidence from the transcripts (to be introduced in §3.1).

3 Our Task

3.1 Task Overview

Our task utilizes LLMs to analyze human conversational transcripts in contact centers. These call transcripts are generated by our in-house Automatic Speech Recognition (ASR) engine. Contact center supervisors can then set up a collection of N questions[1] (defined as *group size* N) about the conversations in the contact center and for each question, a "yes" or "no" judgment is expected (Judgment in the response). Our current setup of considering only binary Yes/No responses is based on the feedback from our clients, thus reflecting the actual demand from the market. We provide some example questions in Appendix A.1.

To enhance the explainability and verifiability of each response and facilitate human review for potential hallucinations, we further introduce an additional Navigation component in the generated responses. Specifically, a transcript is segmented into M utterances,[2] each of which is assigned an index starting from 1 ($m \in \{1, 2, ..., M\}$). Beyond the binary judgment ("yes" and "no") discussed above, LLMs are also instructed to provide one index of the utterance that best supports this judgment. An overview of our task is in Figure 1.

3.2 Structured Output

In our pipeline, we directly prompt LLMs: N questions and the transcript are provided as the context in the prompt, and models are tasked with generating the corresponding N responses for each of the questions.

As multiple answers are retrieved in a single LLM inference pass, in order to generate robust responses in an industrial setting, we develop a JSON-based response structure to include all requested information (Judgment and Navigation), and is reinforced by an in-context formatting example that instructs the LLMs to generate such structure reliably. The template of our designed prompt and the expected response structure are provided in Appendix A.2.

[1]Each question is treated with equal importance, and the ordering of the questions is randomly selected.

[2]An utterance starts when speech begins and ends when a significant pause occurs. Speech from one single person could be split into several utterances and our ASR engine automatically determines where to make these splits.

Percentile	# of Tokens
25th	501
50th	1,153
75th	2,754
95th	6,584

Table 1: Statistics on number of tokens of our internal transcript data, with *tiktoken*[3] used as the tokenizer.

group size N	Judgment Acc	Navigation Acc
5	0.96	0.90
10	0.95	0.91
20	0.91	0.86
30	0.88	0.85
40	0.84	0.81
50	0.84	0.75

Table 2: Human evaluation on Judgment and Navigation accuracy of GPT-4o, with varying group size N. An inter-annotator agreement of 0.85 is achieved for this annotation.

4 Experiments

4.1 Dataset Development

Data Collection. We collect a total of 2,800 conversation transcripts produced by our ASR engine from various contact centers. To better understand the distribution of transcript lengths from these calls, we calculate the number of tokens for different percentiles in Table 1. We notice that more than 25% of our transcripts exceed 2,700 tokens, with notably 5% of the total transcripts extending to as many as 6,500 tokens.

Annotation. Given the long-context nature of our transcript that discussed above, it is impractical to conduct large-scale human annotations for generating answers to all the questions paired with these transcripts.

Since GPT-4o (OpenAI, 2024a) is broadly recognized as one of the top-performing LLMs across various NLP tasks, to address the annotation chal-

[3]The tiktoken tokenizer can be accessed at: https://github.com/openai/tiktoken

lenges, we did an initial round of human evaluation on GPT-4o responses to check whether GPT-4o is a strong and reliable baseline for this task. We present the human evaluation on GPT-4o responses over 50 call transcripts in Table 2. We observe that GPT-4o excels in accuracy with a group size of 5 or 10, both achieving high accuracy close to 0.95 and 0.90 for Judgment and Navigation respectively. However, it presents noticeable decreasing performance for both Judgment and Navigation when the group size becomes larger. Thus, in this study, we group all questions into sets of 10 and use the resulting responses generated by GPT-4o as the standard reference for training and evaluating other models. In other words, we divide 50 questions that are paired with each transcript into five folds, resulting in five rounds of API calls (each containing 10 different questions). In this way, we are able to achieve a balance between maintaining annotation accuracy and reducing the LLM API cost.

Train/dev/test Splits. We sample 2,000, 500 and 300 call transcripts to serve as our training, testing and development sets, respectively.

The goal of this study is to analyze the quality of the generated responses from different models, with different numbers of questions grouped in the prompt (i.e. group size N defined in §3.1). Thus, when compiling the test and development sets, for each group size $N \in \{10, 20, 30, 40, 50\}$, we randomly select N question-answer pairs to pair with each transcript (i.e., there are $300 \times 10 = 3,000$ questions in the test set when the group size N is set to 10). This enables a granular evaluation of model performance when the group size varies in the prompt.

We also explore the use of public LLMs fine-tuned with transcripts associated with N question-answer pairs (denoted by a suffix following the model name such as `Llama3.2-1B-10`, detailed in Table 3 and Figure 2). However, in our pilot studies, we observe that directly applying a fixed group size N in all training data causes various problems, including frequently generating exact N responses irrespective of the instruction, and exhibiting excessive repetition in responses. To mitigate these issues, instead of fixing the group size N, we introduce variability in the training instances by sampling a random number $K \in [5, N]$ and using K question-answer pairs to construct each training instance. For example, in our fine-tuning setup, `Llama3.2-1B-10` means that each transcript in the training set is paired with *up to* 10 question-answer pairs, while the actual number is determined by randomness. Such stochastic approach enhances model robustness and reduces undesirable results in the generated responses.

4.2 Experimental Settings

4.2.1 Models

In this study, we evaluate both proprietary and public models for comprehensive benchmarking.

Proprietary Models. We experiment with three popular commercial models: GPT-4o (OpenAI, 2024a), GPT-4o-mini (OpenAI, 2024a), and Gemini-1.5-flash.[4] GPT-4o is largely perceived as the best LLM for processing text and other modalities (Shahriar et al., 2024), while GPT-4o-mini and Gemini-1.5-flash are the smaller and cheaper options better suited for a production environment.

Llama-3 Models. Llama-3 (Grattafiori et al., 2024) represents the most recent iteration in the Llama series of open-source LLMs and is widely considered as one of the most advanced public models available. Due to the constraints of our industrial model serving environment, we only experiment with two size variations: `Llama3.2-1B` and `Llama3.1-8B`.

For assessing the model's zero-shot capabilities, we utilize the instruction-tuned versions: `Llama3.2-1B-inst` and `Llama3.1-8B-inst`. Conversely, the untuned versions are employed for fine-tuning experiments. Model weights are accessed from Hugging Face (Wolf et al., 2020).

Qwen-2.5 Models. Alibaba's Qwen-2.5 (Yang et al., 2025) is another family of advanced public LLMs, which is engineered to handle a diverse range of tasks with significant advancements in long-text generation and reasoning. In this study, we focus on the `Qwen2.5-7B` version to strike a balance between model performance and scalability. Similar to our setup for Llama-3 models, we select the instruction-tuned version, `Qwen2.5-7B-inst`, for zero-shot evaluations, while the untuned version is reserved for fine-tuning experiments.

4.2.2 Metrics

Judgment Accuracy. For the Judgment portion of the model responses, since it contains only a

[4] https://ai.google.dev/gemini-api/docs/models/gemini

Group size N	Judgment Accuracy (↑)					Navigation F1 (↑)					Navigation MAE (↓)					JSON Decode Error Rate (↓)				
	10	20	30	40	50	10	20	30	40	50	10	20	30	40	50	10	20	30	40	50
gpt-4o	/	**0.90**	**0.89**	**0.87**	**0.85**	/	**0.71**	**0.68**	**0.66**	**0.63**	/	**5.67**	**6.32**	6.81	7.01	0	0	0	0	0
gpt-4o-mini	0.78	0.81	0.82	0.79	0.79	0.58	0.48	0.49	0.48	0.46	6.59	10.13	9.76	9.90	10.42	0	0	0	0	0
gemini-1.5-flash	0.76	0.82	0.81	0.80	0.79	0.57	0.47	0.50	0.52	0.51	5.74	11.76	10.11	10.10	9.86	0	0	0.01	0	0.03
llama3.2-1B-inst	0.58	0.55	0.55	0.53	0.48	0.19	0.16	0.13	0.13	0.11	13.25	14.70	14.98	15.84	16.90	0.48	0.58	0.59	0.62	0.73
llama3.1-8B-inst	0.65	0.61	0.59	0.55	0.53	0.29	0.20	0.21	0.20	0.19	9.40	11.52	12.17	13.12	13.61	0.31	0.39	0.39	0.38	0.49
qwen2.5-7B-inst	0.68	0.60	0.58	0.57	0.52	0.35	0.29	0.30	0.33	0.32	9.74	11.42	11.00	9.64	10.46	0.12	0.23	0.28	0.25	0.43
llama3.2-1B-10	0.68	0.69	0.70	0.68	0.67	0.47	0.39	0.43	0.45	0.44	8.84	11.87	11.45	10.84	10.78	0	0	0	0	0.04
llama3.1-8B-10	**0.86**	0.87	0.88	0.87	0.79	0.63	0.56	0.59	0.60	0.61	5.58	8.38	7.64	7.76	**6.37**	0	0	0	0	0.12
qwen2.5-7B-10	0.81	0.80	0.77	0.75	0.73	**0.66**	0.59	0.61	0.62	0.59	**5.54**	7.96	7.41	**6.59**	7.29	0	0	0.01	0.03	0.01

Table 3: Model evaluation results when varying group size N. The `-10` suffix of fine-tuned Llama and Qwen models denotes up to 10 question-answer pairs are paired with each transcript in the fine-tuning training set. Since we use the generated responses from GPT-4o with N=10 as the standard reference, the corresponding cells are marked with "/". Detailed explanations for the above two settings are provided in §4.1.

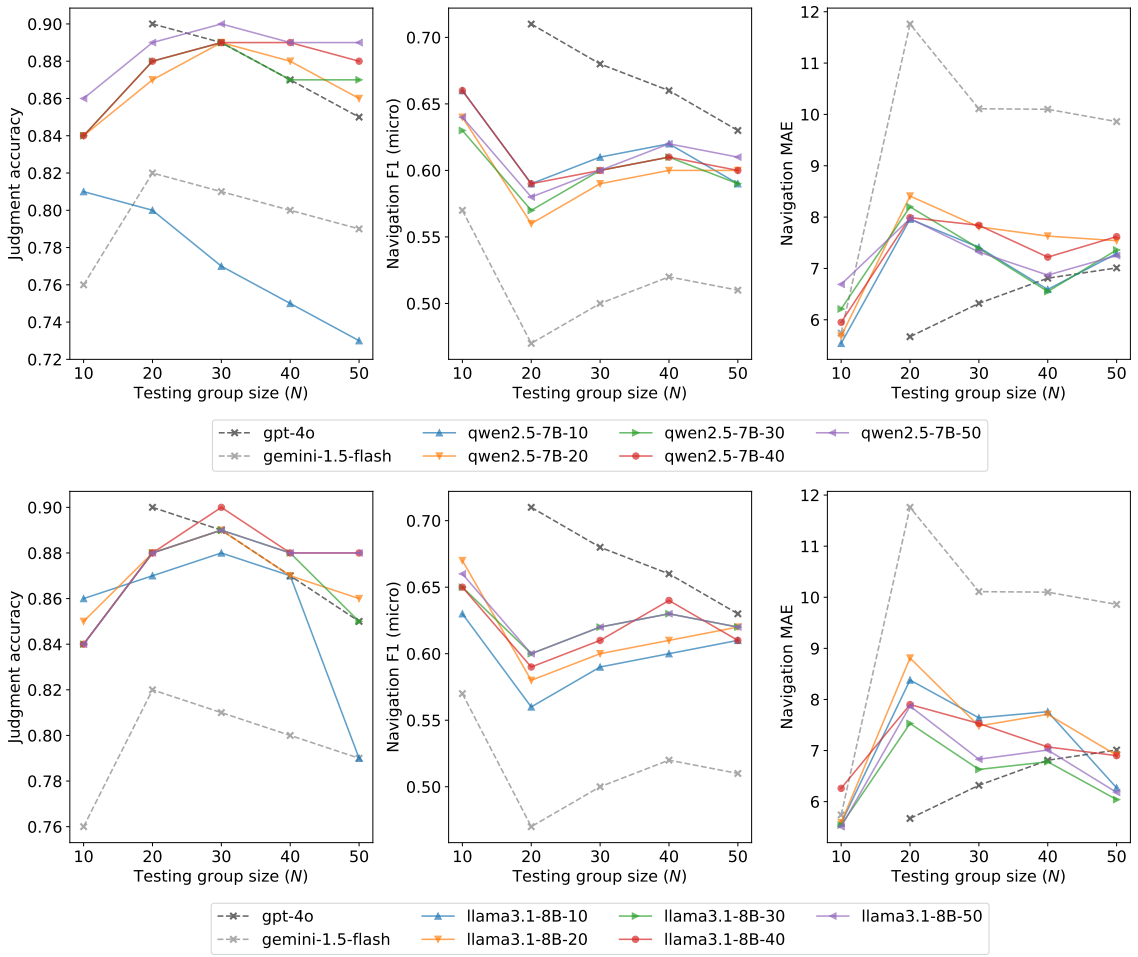

Figure 2: Fine-tuned Qwen2.5-7B (first row) and Llama3.1-8B (second row) evaluation results with varying group size N. The suffix of the model (i.e., `-10`) denotes the *maximum* group size used in fine-tuning (details in §4.1).

binary "yes" or "no" judgment, we directly use accuracy to evaluate these responses.

Navigation F1 and Mean Absolute Error. As detailed in §3.2, the second part of the model responses is Navigation. Since it is a multi-class classification task (i.e., assigning an index from M utterances), we employ an F1 score to measure the model's ability to correctly identify each utterance.

Since information can be spread over multiple conversation turns and several adjacent utterances may be relevant to the topic of a question, we incorporate Mean Absolute Error (MAE) as the additional metric to assess Navigation relevancy.

Specifically, our Navigation MAE is defined as:

$$\text{Navigation MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|\,,$$

where N is the questions' group size, y_i and $\hat{y}_i$ are the true and predicted navigation indices for the i-th prediction, respectively.

4.2.3 Implementation Details

Our subsequent experiments are conducted on a single node with 8 Nvidia A100 GPUs. For all our fine-tuning jobs, we utilize BFloat-16 (Burgess et al., 2019) precision and apply a learning rate of 3e-5 with 3 epochs.

4.3 Results

We vary the group size N for generation and compare the performance of different models on the corresponding test sets. Our experimental results are reported in Table 3. Among all the proprietary models, GPT-4o achieves the best scores across all metrics, while Gemini-1.5-flash ranks the lowest. We observe a decline in performance across all models as the group size N increases, indicating that these LLMs struggle with handling a larger number of queries simultaneously.

Although the instruct versions of the public models cannot deliver equivalent performance as their proprietary counterparts, we find a significant performance increase in those public LLMs after fine-tuning with up to 10 grouped questions, achieving results comparable to — and in some cases surpassing — Gemini-1.5-flash and GPT-4o-mini. A more detailed analysis is provided in §5.1.

5 Analysis

5.1 Fine-Tuning Improves Model Performance

As shown in Table 3, we observe a significant performance increase in small public LLMs after fine-tuning with up to 10 grouped questions (using training data detailed in §4.1). We further explore and compare model performance after more granular fine-tuning with up to N grouped questions, where $N \in \{10, 20, 30, 40, 50\}$, as detailed in §4.1.

Our experimental results are illustrated in Figure 2. It is evident that fine-tuned public LLMs show stronger capabilities in both Judgment and Navigation with more grouped questions included in the training process. For Judgment accuracy, GPT-4o maintains dominant performance with $N \leq 20$; however, fine-tuned Qwen2.5-7B and Llama3.1-8B both surpass GPT-4o with $N \geq 30$. This demonstrates the strong potential of smaller public LLMs. For the Navigation F1 and MAE, GPT-4o still presents the best overall performance, but the gap between public LLMs and GPT-4o narrows noticeably after fine-tuning.

5.2 JSON Decode Error

In our task, answers to multiple questions are generated in a single model output. To enhance the parseability of the output in a production environment, we reinforce a JSON format on our responses (discussed in §3.2). Additionally, we report the error rate of JSON decoding in Table 3 to show the percentage of generated responses that contain any types of JSON decoding issue (key errors, quotation mark mismatches, etc).

While such errors are hardly observed in proprietary LLMs, the instruction-tuned versions of both Llama3 and Qwen2.5 exhibit significant deficiencies in generating correct and desired JSON output structures. However, both these two public models show a substantial reduction in error rates across all group size after fine-tuning with only up to 10 grouped questions.

6 Conclusion

In this study, we evaluate the ability of LLMs to answer multiple questions based on the long conversational context within a single prompt. By adapting batch prompting, we efficiently address the challenge of processing multiple queries with conversational contexts. Through extensive experiments, we observe that while proprietary models like GPT-4o excel in the overall performance, fine-tuned smaller public models like Llama3.1-8B and Qwen2.5-7B can achieve comparable or even superior answer accuracy when more questions are grouped. Our findings demonstrate the viability of fine-tuning smaller models to enhance their capability in processing multiple questions, and underscore the potential for deploying cost-effective models without significant loss of accuracy.

Limitations

Our study has the following limitations:

First, due to several practical constraints, we did not benchmark strong reasoning models emerged recently, such as DeepSeek R1 (DeepSeek-AI, 2025), QwQ (Qwen-Team, 2024) and OpenAI o1

(OpenAI, 2024b), which have demonstrated superior performance in public benchmarks.[5]

Second, due to the nature of our specific QA use case, our experiments focused solely on Yes/No type questions. While our findings may still apply to a broader range of questions, future studies should validate these results using more types of questions.

Third, given the length of transcripts (with more than 25% exceeding 2,700 tokens) and the number of questions paired with each transcript (up to 50 questions), getting high-quality human annotations is practically challenging and costly with our limited resources. Thus, as noted in Section 4.1, we adopt GPT-4o-generated labels after conducting a careful correlation study between human annotators and model-generated labels, which demonstrated strong agreement when $N \leq 20$. However, we acknowledge that in an ideal situation where adequate well-trained annotators are available, a full-scale annotation job will produce more accurate benchmarking datasets.

References

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Neil Burgess, Jelena Milanovic, Nigel Stephens, Konstantinos Monachopoulos, and David Mansell. 2019. Bfloat16 processing for neural networks. In *2019 IEEE 26th Symposium on Computer Arithmetic (ARITH)*, pages 88–91.

Vittorio Castelli, Rishav Chakravarti, Saswati Dana, Anthony Ferritto, Radu Florian, Martin Franz, Dinesh Garg, Dinesh Khandelwal, Scott McCarley, Michael McCawley, Mohamed Nasr, Lin Pan, Cezar Pendus, John Pitrelli, Saurabh Pujar, Salim Roukos, Andrzej Sakrajda, Avi Sil, Rosario Uceda-Sosa, Todd Ward, and Rong Zhang. 2020. The TechQA dataset. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1269–1278, Online. Association for Computational Linguistics.

Jian Chen, Peilin Zhou, Yining Hua, Loh Xin, Kehui Chen, Ziyuan Li, Bing Zhu, and Junwei Liang. 2024. Fintextqa: A dataset for long-form financial question answering. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, page 6025–6047. Association for Computational Linguistics.

[5] https://lmarena.ai/

Zhoujun Cheng, Jungo Kasai, and Tao Yu. 2023. Batch prompting: Efficient inference with large language model APIs. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 792–810, Singapore. Association for Computational Linguistics.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Aaron Grattafiori, Abhimanyu Dubey, and Abhinav Jauhri et al. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2020. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Preprint*, arXiv:2009.13081.

Yunshi Lan, Gaole He, Jinhao Jiang, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. 2023. Complex knowledge base question answering: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 35(11):11196–11215.

Jonathan Li, Rohan Bhambhoria, Samuel Dahan, and Xiaodan Zhu. 2024. Experimenting with legal ai solutions: The case of question-answering for access to justice. *Preprint*, arXiv:2409.07713.

Jianzhe Lin, Maurice Diesendruck, Liang Du, and Robin Abraham. 2024. Batchprompt: Accomplish more with less. In *The Twelfth International Conference on Learning Representations*.

Thomas Mueller, Francesco Piccinno, Peter Shaw, Massimo Nicosia, and Yasemin Altun. 2019. Answering conversational questions on structured data without logical forms. 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)*, pages 5902–5910, Hong Kong, China. Association for Computational Linguistics.

OpenAI. 2024a. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

OpenAI. 2024b. Openai o1 system card. *Preprint*, arXiv:2412.16720.

Qwen-Team. 2024. Qwq: Reflect deeply on the boundaries of the unknown.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. CoQA: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

Sakib Shahriar, Brady Lund, Nishith Reddy Mannuru, Muhammad Arbab Arshad, Kadhim Hayawi, Ravi Varma Kumar Bevara, Aashrith Mannuru, and Laiba Batool. 2024. Putting gpt-4o to the sword: A comprehensive evaluation of language, vision, speech, and multimodal proficiency. *Preprint*, arXiv:2407.09519.

Minzheng Wang, Longze Chen, Fu Cheng, Shengyi Liao, Xinghua Zhang, Bingli Wu, Haiyang Yu, Nan Xu, Lei Zhang, Run Luo, Yunshui Li, Min Yang, Fei Huang, and Yongbin Li. 2024. Leave no document behind: Benchmarking long-context LLMs with extended multi-doc QA. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5627–5646, Miami, Florida, USA. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 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, Online. Association for Computational Linguistics.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Munazza Zaib, Wei Emma Zhang, Quan Z. Sheng, Adnan Mahmood, and Yang Zhang. 2022. Conversational question answering: a survey. *Knowledge and Information Systems*, 64(12):3151–3195.

Fengbin Zhu, Wenqiang Lei, Chao Wang, Jianming Zheng, Soujanya Poria, and Tat-Seng Chua. 2021. Retrieving and reading: A comprehensive survey on open-domain question answering. *Preprint*, arXiv:2101.00774.

A Appendix

A.1 Example Questions

We provide some example Yes/No type questions that can be useful in our customer support domain:

- Did the agent introduce themselves?
- Was the account information verified?
- Was the order confirmed?
- Did the agent ask if further assistance was required?
- Did the agent confirm the customer's preferred communication channel?

A.2 Prompt Template

Our prompt template is as follows:

```
Here is a conversation transcript:

            Utterance 1:
            Utterance 2:
                 ...
            Utterance M:

Answer the following questions based
on the above conversation transcript:

             Question 1:
             Question 2:
             Question 3:
                 ...
             Question N:

The result should be in the JSON format,
use the index of each question as the
key, while the value should be an array
consists of two parts:  the first part
is a string starting with a "Yes" or
"No" answer to the question followed by
justification.  For the second part, re-
turn the index of one utterance from the
transcript that can best support your
justification, for "No" answers just use
"NA" as the second part.  Do not provide
any judgment on conversation quality.
An example of the output format:

{
    "Q1": ["Yes, the agent verified customer's
    information at the start of the call", "5"],
    "Q2": ["No, the agent did not send a copy
    to the customer.", "NA"]
}
```