MEETINGQA2: A Generative Speaker-Centric Benchmark for Multi-Party Dialogue Question Answering

Anonymous ACL submission

Abstract

Multi-party Dialogue Question Answering (MDQA) requires the machine to answer questions based on a conversation with multiple interlocutors. Recent progress in MDQA has exposed critical limitations in existing datasets, including extractive answer constraints, single-answer assumptions, and limited evaluation of holistic dialogue understanding. While MEETINGQA (Prasad et al., 2023) introduces multi-span and multi-speaker answers, it inherits answer positional biases from utterance-derived questions and restricts answers to extractive formats. To address these limitations, we introduce MEETINGQA2, the first generative Question Answering (QA) dataset for multi-party dialogues, featuring speaker-centric summary-style answers and generated QA pairs through Large Language Model (LLM) prompting. It comprises 1,033 QA pairs, including 108 unanswerable questions, to rigorously assess dialogue comprehension. Experimental evaluations demonstrate substantial performance gaps in LLMs, revealing limitations in understanding multi-party dialogue contexts.

1 Introduction

Recent advancements in conversational AI have spurred a growing interest in multi-party dialogue research, particularly in QA systems (She et al., 2024; Li et al., 2023). While existing datasets such as Molweni and FriendsQA provide data benchmarks for MDQA, they exhibit two critical limitations. First, their extractive nature confines answers to short text spans within narrowly constrained dialogue contexts, typically averaging fewer than 400 words. Second, these datasets assume single-answer scenarios and focus on simple entity and noun phrase answers. They overlook the inherent complexity of real-world discussions, in which multiple speakers often express divergent perspectives. In this case, meaningful answers often span multiple utterances and involve contributions from more than one speaker.

The most relevant work, MEETINGQA (Prasad et al., 2023), attempts to address these limitations by constructing QA pairs from annotated AMI meeting transcripts (Carletta et al., 2005). While it permits multi-speaker and multi-span answers, two fundamental constraints remain: (1) The direct extraction of questions from dialogue utterances introduces positional biases, enabling models to exploit sequential patterns (e.g., leveraging adjacent utterances as answers due to sequential dependencies) rather than demonstrating genuine comprehension. (2) Answer formats remain strictly extractive, which fails to assess models' ability to synthesize information across long conversational contexts.

To bridge this research gap, we present MEETINGQA2, a novel dataset derived from the MEETINGQA test set. Unlike other datasets, MEETINGQA2 is a generative MDQA dataset where questions are reconstructed by GPT-4o[1] and answers are in the form of summary-style and speaker-centric. These answers clearly distinguish each speaker's views and summarize the relevant context. In Figure 1, the reformatted answer presents each speaker's perspective (e.g., Speaker0 and Speaker2) in separate sections. This organization helps users quickly identify and comprehend different viewpoints, improving answer clarity and usability. MEETINGQA2 is constructed via a two-stage methodology: (1) Question reformulation to generate natural questions from original QA pairs. (2) Answer reformulation based on the reformulated questions. It features extensive multi-party conversations, with the longest dialogue containing 19,225 words from an extended group discussion. The final collection comprises 1,033 QA pairs, including 108 unanswerable questions.

Our task setting is challenging and interesting in

[1] https://openai.com/index/hello-gpt-4o/

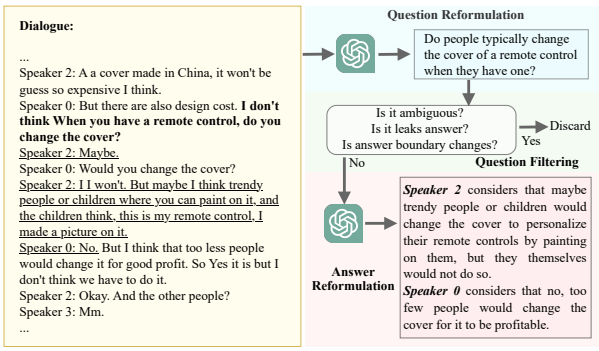

Figure 1: Data construction process for answerable cases in MEETINGQA2. The bold and underlined sentences in the yellow box represent the questions and corresponding answers from MEETINGQA, respectively.

several ways. Meeting transcripts are long, requiring models to reason over extended contexts (Pang et al., 2022; Soleimani et al., 2021). Answering accurately further requires both precise localization of relevant dialogue segments and deeper discourse-level understanding. In addition, speaker-centric summary-style answers can better represent the dialogue understanding capabilities of the models.

The contributions of our work include: 1) To the best of our knowledge, MEETINGQA2 is the first generative MDQA dataset. We offer a high-quality and challenging evaluation benchmark where questions are generated from utterances within the conversation, and answers are summary-style centered around speakers. 2) We evaluate the performance of LLMs on MEETINGQA2 and conduct error analysis. Results indicate substantial room for improvement in the multi-party dialogue comprehension capabilities of existing models, such as speaker identification and factual understanding.

2 Our Dataset: MEETINGQA2

2.1 Data Construction

As shown in Figure 1, our dataset creation framework employs a hybrid approach combining LLM-powered rewriting with multi-stage quality assurance. It consists of three phases for answerable questions: question reformulation, question filtering, and answer reformulation.

To avoid potential complications from long contexts during the data generation phase and to ensure high-quality and grounded QA pairs, we leverage the short dialogue context version provided by MEETINGQA for our dataset construction. However, it is crucial to note that during the final evaluation, models are tasked with answering these questions using the complete and full-length dialogue transcript, thereby directly assessing their capabilities in long-context comprehension.

2.1.1 Question Reformulation

Building upon the extractive QA framework of MEETINGQA, we first convert extracted questions Q into their non-extracted counterparts Q'. Specifically, to enhance GPT's ability to resolve referential and elliptical expressions, we first prepend 20 preceding dialogue turns to each sample. To avoid contamination from post-question context, we place the source utterance of Q as the final turn in the input sequence. The resulting input consists of three components: (1) the expanded dialogue context, (2) the original question Q, and (3) the associated answer span(s) A. This enriched context is subsequently formatted into a structured prompt for few-shot question generation using GPT-4o. The model is explicitly instructed to incorporate referential and elliptical cues when reformulating questions. Complete prompt details appear in Table 4.

2.1.2 Question Filtering

To ensure high quality of questions, we develop a multi-stage filtering system to address the issues of ambiguity resolution, answer leakage, and answer boundary changes.

Ambiguity Resolution Detection Even after question reformulation, some questions remain semantically vague (e.g., the original question "*Or are you going to prime them in any way?*" is transformed into "*Are you planning to influence the participants with the language choice or content of the voice-over in any way?*", retaining ambiguity "*you*"). To address this, we exclude any reformulated question Q' containing ungrounded pronouns (e.g., "*he*", "*you*") without explicit speaker attribution "*Speaker X*". This step retents 76.6% samples.

Answer Leakage Detection In the question reformulation process, answer spans A are input into GPT-4o to generate Q', which may inadvertently reveal the answers. To mitigate this risk, we discard samples where the ROUGE-L recall between Q' and A is ≥ 0.4, preserving 80.9% of the dataset. The effects of alternative leakage detection strategies are detailed in Appendix C.

Answer Boundary Detection In MEETINGQA, questions are constrained to the ongoing discussion topic, which limits answers to a specific topical scope. In contrast, our reformulated questions may broaden the answer

scope beyond the original spans by referencing the conversation as a whole. Detailed explanations are shown in the Appendix D.

To address this issue, we implement a two-stage GPT-assisted method to filter these cases. First, we perform backward context extension by 50 consecutive utterances. Second, we instruct GPT-4o to localize answer boundaries by identifying specific utterance IDs within the extended context, so as to identify samples with a larger range of answers. The prompt is shown in Table 5. This automated screening identifies 189 potentially over-extended samples from the initial 1,061 candidates. Subsequently, all flagged instances underwent manual verification through expert review, which confirms 136 cases as genuine overextension cases requiring removal from the final dataset.

2.1.3 Answer Reformulation

In the final stage, GPT-4o generates speaker-centric summary-style answers A' through structured prompt engineering. Specifically, the input for GPT includes: (1) the expanded dialogue comprising 20 preceding conversational turns, (2) the reformulated question Q' and (3) the original answer span A. They are formatted into a highly structured prompt to require the LLM to generate answers ordered by speaker, as detailed in Table 6.

2.1.4 Unanswerable Question Construction

Our approach to constructing unanswerable questions slightly differs from that for answerable ones. Specifically, it comprises three primary steps. First, we filter out ambiguous questions that often lead to inconsistent or fabricated outputs during GPT generation. In particular, questions with fewer than two meaningful words after stopword removal are discarded. Second, we reformulate the remaining questions using GPT-4o as described in Section 2.1.1. Third, we use GPT-4o to exclude questions that become answerable after reformulation. The prompt is shown in Table 7. Finally, we obtain 108 rigorously unanswerable question samples.

2.2 Statistics and Analysis

Table 1 presents a comparison between MEETINGQA and MEETINGQA2. Both datasets exhibit similar dialogue complexity, with an average of 8,755 and 8,741 words per dialogue, respectively. However, differences emerge in their QA formulations. MEETINGQA2 features a 51.7% increase in average question length (from 11.6 to 17.6 words) and a 15.1% increase in average answer length (from 40.5 to 46.6 words), with reduced maximum lengths for both. These structural changes stem from the reformulation process, which includes coreference resolution to expand average textual length, speaker-centric summarization to condense overly long answers, and free-form restructuring to normalize question phrasing. Despite these modifications, both datasets maintain comparable proportions of multi-span and multi-speaker answers, ensuring the core linguistic challenges remain intact during reformulation. In addition, the dataset size reflects a deliberate trade-off between quantity and quality, prioritizing more reliable reformulations while keeping the overall scale sufficient for meaningful evaluation. The quality verification of MEETINGQA2 shows in Appendix B.

	MEETINGQA	MEETINGQA2
Formulation	Span	Free-form
Question count	2,467	1,033
Avg/Max dialog length	8,755.7/19,225	8,741.2/19,225
Avg/Max question length	11.6/155	17.6/41
Avg/Max answer length	40.5/228	46.6/144
Multi-span answer ratio	38.7%	38.6%
Multi-speaker answer ratio	49.1%	48.1%

Table 1: Comparison between test sets of MEETINGQA and MEETINGQA2.

3 Experimentation

3.1 Settings

Our experimental study evaluates generative LLMs from two architectural families: Meta's LLaMA3 series (Dubey et al., 2024), OpenAI's GPT-3.5 Turbo[2] and GPT-4o[1]. We design three distinct experimental protocols to systematically assess their performance in MDQA scenarios. They are Zero-Shot Inference (ZSI), Retrieval-Augmented Generation (RAG), and Standard Context Answer Generation (SCG), respectively. Detailed information is shown in Appendix E.

The evaluation metrics we employ are BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), BERTScore (Zhang et al., 2019) and MoverScore (Zhao et al., 2019). Detailed information appears in Appendix E.1.

3.2 Main Results

Table 2 shows the performance of various LLMs on MEETINGQA2. Notably, even under the SCG

[2] https://openai.com/index/gpt-3-5-turbo-fine-tuning-and-api-updates/

Model	B4	RL	BS	MS
		ZSI		
LLaMA3.2-3B	8.83	26.08	26.05	20.63
LLaMA3-8B	12.24	29.51	29.98	22.03
LLaMA3.1-8B	**15.96**	**35.10**	**37.26**	**26.30**
GPT-3.5 Turbo	7.81	21.01	21.13	16.94
GPT-4o	11.32	27.99	30.86	24.03
		RAG		
LLaMA3.2-3B	12.25	30.01	31.19	23.51
LLaMA3-8B	**17.35**	**35.16**	**37.23**	**27.44**
LLaMA3.1-8B	16.84	33.68	35.58	26.05
GPT-3.5 Turbo	9.48	23.91	26.07	20.66
GPT-4o	12.85	28.94	32.23	24.84
		SCG		
LLaMA3.2-3B	15.57	33.14	34.54	25.81
LLaMA3-8B	18.59	**36.59**	38.67	**28.85**
LLaMA3.1-8B	**19.02**	36.43	**38.73**	28.51
GPT-3.5 Turbo	10.99	25.17	28.41	22.19
GPT-4o	15.79	32.73	36.60	28.28

Table 2: Performance on MEETINGQA2 under different experimental configurations (B4: BLEU-4, RL: ROUGE-L, BS: BERTScore, MS: MoverScore).

setting, the best results, 36.59% of ROUGE-L and 38.73% of BERTScore, remain relatively modest. These results underscore the persistent challenges LLMs face in understanding the complex dynamics of multi-party discourse. We highlight several key observations:

First, both model scale and pretraining strategy profoundly affect RAG efficacy. Interestingly, while the long-text-pretrained LLaMA3.1-8B shows marginal gains from RAG over zero-shot performance, smaller models benefit substantially. For instance, LLaMA3.2-3B improves by 5.14% in BERTScore (from 26.05% to 31.19%), and LLaMA3-8B, which lacks long-text pretraining, sees an even greater boost of 7.25% (from 29.98% to 37.23%). This trend suggests that larger models with long-text pretraining may already internalize robust reasoning capabilities, reducing their reliance on external retrieval.

Second, retrieval quality remains a critical bottleneck for RAG. Across the board, model performance improves when provided with oracle (ground-truth) context, which indicates that current retrieval systems often miss key information necessary for accurate answering. It highlights the importance of improving retrieval precision in multi-party settings, where relevant information is sparse and distributed.

Third, in the zero-shot setting, LLaMA3.1-8B, which has been trained on long-text data, achieves the best performance among open-source models. In contrast, LLaMA3-8B exhibits a significant performance gap, underscoring the advantage of models directly pretrained on long-text over those relying on length extrapolation techniques.

Finally, we observe that GPT shows suboptimal performance under evaluation metrics. This primarily arises from its propensity to generate contextually appropriate yet verbose responses, frequently incorporating supplementary content, as shown in Figure 2. These answers are often penalized by metrics such as BLEU-4. This phenomenon emphasizes the need for metrics that prioritize the assessment of answer correctness and contextual appropriateness, rather than relying solely on lexical overlap or similar phrase matching.

3.3 Error Analysis

We conduct a manual evaluation of 50 randomly sampled answers generated by LLaMA3.1-8B, and observe several notable findings. Note that a sample may exhibit multiple types of errors. Specifically, (1) Only 16% of the answers are fully correct, primarily addressing simpler yes/no verification questions involving single-speaker or single-span content. (2) Speaker-related errors are the most common, occurring in 40% of cases, of these, 90% involve speaker omissions and 50% feature speaker misattributions, which highlights significant weaknesses in speaker identification. (3) Factual inaccuracies are present in 24% of answers, which indicates difficulties in understanding complex conversational contexts. (4) 10% of answers are incomplete, omitting essential information; 6% are entirely irrelevant, suggesting comprehension failures; and 4% inappropriately attempt to answer unanswerable questions. These findings highlight critical limitations in current LLMs' ability to handle multi-party dialogue, especially in tracking speaker roles, maintaining factual consistency, and recognizing question answerability.

4 Conclusion

We present MEETINGQA2, an MDQA dataset featuring speaker-centric, summary-style answers and LLM-generated QA pairs. Most QA pairs involve multiple utterances or speakers, reflecting the complexity of real-world multi-party interactions. Experimental results show that current LLMs struggle significantly with such intricate dialogue dynamics, making MEETINGQA2 a valuable benchmark for dialogue understanding.

Limitations

While MEETINGQA2 establishes a novel benchmark in MDQA systems, it exhibits several notable constraints that warrant attention. Primarily, the dataset's limited scale restricts its capacity to support extended training periods essential for developing robust neural models. This scale is partly a consequence of our design choice to prioritize benchmark reliability over sheer quantity: each QA pair is produced via a multi-stage pipeline that requires multiple LLM invocations and stringent filtering (e.g., reformulation, ambiguity filtering, answer leakage detection, boundary detection, and generative answer summarization), which is computationally and financially costly. As a result, we focus on constructing a smaller set of dense, complex, and carefully verified QA pairs. Moreover, our dialogue sources solely focus on meeting transcripts, overlooking other types of conversational transcripts like forum discussions, TV series transcripts, etc., potentially constraining the utility of MEETINGQA2. Addressing these limitations requires future research in the field of MDQA.

References

Jean Carletta, Simone Ashby, Sebastien Bourban, Mike Flynn, Mael Guillemot, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, Wessel Kraaij, Melissa Kronenthal, et al. 2005. The ami meeting corpus: A pre-announcement. In *International workshop on machine learning for multimodal interaction*, pages 28–39.

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. *arXiv preprint arXiv:2407.21783*.

Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. A survey on rag meeting llms: Towards retrieval-augmented large language models. In *Proceedings of the SIGKDD 2024*, pages 6491–6501.

Sarah E. Finch and Jinho D. Choi. 2024. ConvoSense: Overcoming monotonous commonsense inferences for conversational AI. *Transactions of the Association for Computational Linguistics*, pages 467–483.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Deepanway Ghosal, Siqi Shen, Navonil Majumder, Rada Mihalcea, and Soujanya Poria. 2022. CICERO: A dataset for contextualized commonsense inference in dialogues. In *Proceedings of the ACL 2022*, pages 5010–5028.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT 2019*.

Hengli Li, Song-Chun Zhu, and Zilong Zheng. 2023. Diplomat: a dialogue dataset for situated pragmatic reasoning. volume 36, pages 46856–46884.

Jiaqi Li, Ming Liu, Min-Yen Kan, Zihao Zheng, Zekun Wang, Wenqiang Lei, Ting Liu, and Bing Qin. 2020. Molweni: A challenge multiparty dialogues-based machine reading comprehension dataset with discourse structure. In *Proceedings of the COLING 2020*, pages 2642–2652.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Proceedings of the ACL 2004*, pages 74–81.

Richard Yuanzhe Pang, Alicia Parrish, Nitish Joshi, Nikita Nangia, Jason Phang, Angelica Chen, Vishakh Padmakumar, Johnny Ma, Jana Thompson, He He, and Samuel Bowman. 2022. QuALITY: Question answering with long input texts, yes! In *Proceedings of the NAACL 2022*, pages 5336–5358.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the ACL 2002*, pages 311–318.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. 2024. Yarn: Efficient context window extension of large language models. In *Proceedings of the ICLR 2024*, pages 1–20.

Archiki Prasad, Trung Bui, Seunghyun Yoon, Hanieh Deilamsalehy, Franck Dernoncourt, and Mohit Bansal. 2023. MeetingQA: Extractive question-answering on meeting transcripts. In *Proceedings of the ACL 2023*, pages 15000–15025.

Shuaijie She, Shujian Huang, Xingyun Wang, Yanke Zhou, and Jiajun Chen. 2024. Exploring the factual consistency in dialogue comprehension of large language models. In *Proceedings of the NAACL 2024*, pages 6087–6100.

Siqi Shen, Deepanway Ghosal, Navonil Majumder, Henry Lim, Rada Mihalcea, and Soujanya Poria. 2022. Multiview contextual commonsense inference: A new dataset and task. *arXiv preprint arXiv:2210.02890*.

Amir Soleimani, Christof Monz, and Marcel Worring. 2021. NLQuAD: A non-factoid long question answering data set. In *Proceedings of the EACL 2021*, pages 1245–1255.

Kai Sun, Dian Yu, Jianshu Chen, Dong Yu, Yejin Choi, and Claire Cardie. 2019. DREAM: A challenge data set and models for dialogue-based reading comprehension. *Transactions of the Association for Computational Linguistics*, pages 217–231.

Xindi Wang, Mahsa Salmani, Parsa Omidi, Xiangyu Ren, Mehdi Rezagholizadeh, and Armaghan Eshaghi. 2024. Beyond the limits: A survey of techniques to extend the context length in large language models. *arXiv preprint arXiv:2402.02244*.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. 2024. C-pack: Packed resources for general chinese embeddings. In *Proceedings of the SIGIR 2024*, pages 641–649.

Peng Xu, Wei Ping, Xianchao Wu, Lawrence McAfee, Chen Zhu, Zihan Liu, Sandeep Subramanian, Evelina Bakhturina, Mohammad Shoeybi, and Bryan Catanzaro. 2024. Retrieval meets long context large language models. In *Proceedings of the ICLR 2024*, pages 1–16.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Zhengzhe Yang and Jinho D Choi. 2019. FriendsQA: Open-domain question answering on tv show transcripts. In *Proceedings of the SIGDIAL 2019*, pages 188–197.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M Meyer, and Steffen Eger. 2019. Moverscore: Text generation evaluating with contextualized embeddings and earth mover distance. *arXiv preprint arXiv:1909.02622*.

A Related Work

A.1 Multi-party Dialogue Question Answering Datasets

Recent years have witnessed the emergence of MDQA datasets, primarily sourced from two domains: chit-chat exchanges in TV shows (Sun et al., 2019; Yang and Choi, 2019) and dialogues in the Ubuntu Chat Corpus (Li et al., 2020). However, these works suffer from notable limitations: (1) Human-generated questions often lead to shallow queries with brief, single-speaker answers. (2) Dialogue contexts are typically short, averaging under 400 words. Other dialogue QA datasets construct generative QA pairs using manual annotations (Ghosal et al., 2022; Shen et al., 2022) or GPT (Finch and Choi, 2024), but their dialogues are limited to two-person conversations. MEETINGQA (Prasad et al., 2023) attempts to address these limitations by constructing QA pairs from meeting transcripts (Carletta et al., 2005), allowing for multi-speaker and multi-span answers. However, it still has constraints: (1) Questions are directly derived from dialogue utterances, creating positional biases that allow models to rely on sequential patterns instead of showcasing true understanding. (2) Answers are strictly extractive, limiting the assessment of models' ability to synthesize information.

Our work builds upon the MEETINGQA foundation but extends it to a generative QA setting, enabling more comprehensive evaluation of LLMs' conversational understanding capabilities.

A.2 Question Answering on Long Context

Contemporary studies reveal that QA models exhibit limitations in accurately responding to questions requiring comprehension of extended contextual information (Pang et al., 2022; Soleimani et al., 2021). To address this challenge, researchers have developed two primary mitigation strategies. The first approach employs RAG frameworks (Gao et al., 2023; Xu et al., 2024; Fan et al., 2024), which implement a two-stage process involving relevant paragraph retrieval followed by context-aware answer generation. The second solution leverages parameter-efficient length extrapolation techniques (Peng et al., 2024; Wang et al., 2024), with empirical evidence demonstrating significant improvements in model performance on long-text comprehension tasks. Moreover, contemporary LLMs are increasingly integrating progressive context window expansion during their pre-training phases to bolster their capabilities in long-context reasoning. This strategy is used by prominent models including LLaMA (Dubey et al., 2024) and Qwen (Yang et al., 2024), which systematically increase their context window sizes through scaled training methodologies.

In our experiments, we utilize both RAG frameworks and length extrapolation techniques, specifically dynamic RoPE scaling, to evaluate their impact on performance. Additionally, we employ LLMs that are adept at handling varying lengths of text to ensure comprehensive analysis.

B Quality Verification

To evaluate the quality of the GPT-generated QA pairs, we manually inspect 300 randomly sampled instances (approximately 30% of the dataset). Three expert annotators focus on evaluating the factual correctness and consistency of each QA pair. This verification process achieves a 95% accuracy rate, indicating that the majority of the GPT-generated QA pairs are highly accurate and reliable for practical use.

We further conduct a qualitative error analysis of the remaining 5% incorrect cases and categorize them into the following types: a) factual hallucination (1%), where the answer introduces unsupported claims not grounded in the source; b) minor content omission (1%), where a relevant speaker-related detail is partially missing, leading to a slightly incomplete response; and c) stylistic awkwardness (3%), where the QA content is factually plausible but exhibit unnatural or awkward phrasing (e.g., overly rigid formulations), reducing fluency and readability. These findings suggest that most errors are attributable to surface-level expression rather than substantive factual inconsistency, while hallucinations and omission errors occur relatively rarely in our dataset.

C Impact of Different Answer Leakage Detection Methods

To ensure the quality of question formulation, we investigate two distinct methods for identifying and eliminating questions with answer leakage. The first approach (**TBF: Threshold-based Filtering**) establishes a rigorous criterion by discarding samples exhibiting ROUGE-L recall ≥ 0.4 between the reformulation question Q' and original answer spans A. The second strategy (**LAV: LLM-assisted Validation**) implements a filtering mechanism through a task-specific few-shot prompting strategy with GPT-4. The complete prompt structure is detailed as Table 8.

We conduct a comparative evaluation using 100 randomly sampled instances to assess detection effectiveness. Table 3 presents the performance metrics, demonstrating our ultimate implementation of the threshold-based approach.

	TBF	LAV
Recall	100%	80%
Precision	29.41%	12.12%

Table 3: Comparative performance evaluation of answer leakage detection methods.

D Supplementary Explanations on Answer Boundary Detection

The motivation for performing answer boundary detection stems from the latent assumption of extractive QA pairs in MEETINGQA. In MEETINGQA, questions inherently relate to the ongoing discussion topic, naturally confining answers within that scope. However, reformulated questions may broaden the answer scope beyond the original spans by targeting the entirety of a lengthy conversation.

For instance, the original question "*Now have we decided anything, are we doing anything?*" is a sentence within the dialogue and carries positional context, it pertains specifically to a certain topic. In contrast, the reformulated question "*Have Speaker 3 and Speaker 4 made any decisions or decided on any actions to take?*" is a free-form question without dialogue location information. It is for all discussion topics. Consequently, the scope of possible answers shifts accordingly.

E Additional Experimental Settings

Zero-Shot Inference (ZSI) In this configuration, models generate answers directly from dialogue and question. Detailed prompt is shown in Table 9. For long context constrained models like LLaMA3-8B, we use dynamic RoPE scaling[3] for sequence length extrapolation. For GPT-3.5 Turbo suited for tokens up to 16k, we post-truncate tokens exceeding 16k. Remaining models employ standard zero-shot inference.

Retrieval-Augmented Generation (RAG) Given the extensive length of meeting transcripts, a suitable solution is RAG (Fan et al., 2024; Gao et al., 2023). The method involves three steps, First, each dialogue is divided into 512-token segments using sliding windows with 5-utterance overlaps, preserving conversational coherence while averting data fragmentation. Second, we utilize the BGE-large retriever (Xiao et al., 2024) to compute the dot product between query embeddings and pre-indexed chunks, returning top-3 relevant chunks. This step achieves 77.8% recall on MEETINGQA2. Finally, the retrieved chunks are combined with the question to facilitate

[3] https://www.reddit.com/r/LocalLLaMA/comments/14mrgpr/dynamically_scaled_rope_further_increases/

answer generation by models.

Standard Context Answer Generation (SCG) In this setting, the context from the answer reformulation stage is provided to eliminate retrieval bias, simulating a 100% recall scenario to isolate model performance from retrieval limitations.

E.1 Metrics

The evaluation criteria we employ can be categorized as follows: (1) N-gram Metrics: BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004) measure lexical overlap between generated and reference texts. (2) Semantic Metrics: BERTScore (Zhang et al., 2019) and MoverScore (Zhao et al., 2019). BERTScore leverages BERT's (Kenton and Toutanova, 2019) contextual embeddings to capture semantic similarity, and we rescale it to enhance interpretability[4]. MoverScore extends BERTScore through optimal transport theory, enabling soft many-to-one word alignments by minimizing transformation costs between text pairs.

F Prompts

The prompts we use in our method are shown in Table 4 5 6 7.

Given the dialogue a question and answers, you need to rewrite the question to generate a complete and clear question to the answer. There should be no ambiguity or omission in the rewritten question. If pronouns appear in the question, they need to refer back to the correct entities. Let's think step by step. Output the generated content as follows: # Thinking Process: xxx # Rewrite Question: xxx

Table 4: Question reformulation prompt.

Give the dialogue, a question, you need to give utterances that can answer the question. Only output the utterance ids of utterances that can answer the question.

Table 5: Answer boundary detection prompt.

Given the dialogue, a question and answers of corresponding speakers, you need to generate summaries of the answers. There should be no ambiguity or omission in the summaries. If a summary can answer the question directly, it needs to answer the question first, then make a supplement if have supplementary information. The summary content must come from the specific speaker who answers. Let's think step by step. Output the generated content in JSON format as follows. "Answer-related speaker": List the speakers who give the answers, "Speaker 1": Speaker 1 considers that ... (the summary of speaker 1's answer), "Speaker 2": Speaker 2 considers that ... (the summary of speaker 2's answer)

Table 6: Answer reformulation prompt.

You need to determine whether the dialogue can answer the given question. Let's think step by step. Output the generated content as follows. # Thinking Process: The thinking process of determining the answerability of the question. # Answerability: If the question is answerable, output 'Yes', otherwise output 'No'.

Table 7: Answerability detection prompt.

Give the dialogue, a question and answers, you need to judge if the question leaks the answer. Output the generated content as follows: # Thinking Process: The thinking process of the leakibility of the question. # Leak or not: Yes or No.

Table 8: Answer leakage detection prompt.

Given the dialogue and a question, you need to generate the answer. The answer may involve multiple utterances or speakers. The answer can't just be a dialogue span. You need to summarize the views of the speakers who answered the question, respectively. You should generate content as follows: Speaker x considers that ...

Table 9: Zero-shot prompt.

[4] https://github.com/Tiiiger/bert_score/blob/master/journal/rescale_baseline.md

Dialogue:

Speaker 1: NITE object model, I think.
Speaker 2: NITE.
Speaker 3: Okay, that makes sense.
Speaker 0: What do you say?
Speaker 1: NITE object model, I think that's With the removing the disfluencies, have you done anything with that yet?
Speaker 0: Okay.
Speaker 3: Mm.
Speaker 1: Anybody.
Speaker 3: No.
Speaker 0: No, we haven't.
Speaker 1: Cause I was wondering how we're gonna write it back to XML, if we did the processing.
Speaker 3: I thought we were just going to have that as words.
Speaker 1: Writing it back to XML Seems There's a blop of text.

Question: Have Speaker 3 and Speaker 0 done anything with removing disfluencies using the NITE object model yet?

Gold: Speaker 3 considers that they have not done anything with removing disfluencies using the NITE object model yet. Speaker 0 confirms that they have not done anything with removing disfluencies using the NITE object model yet.

LLaMA3.1-8B (BS=0.51): Speaker 3 considers that they haven't done anything with removing disfluencies using the NITE object model yet, as they say "No" to the question "With the removing the disfluencies, have you done anything with that yet?".

GPT-3.5 Turbo (BS=0.28): Speaker 3 and Speaker 0 have not done anything with removing disfluencies using the NITE object model yet. They discussed the possibility of removing disfluencies and displaying the text without disfluencies in a text or XML format. They mentioned the option of using XSLT to transform the text into XML. However, they also considered the option of selectively printing the text without disfluencies on the fly. They acknowledged that displaying the text without disfluencies would limit the ability to search and listen to the audio. Overall, they concluded that they would focus on displaying the text without disfluencies as a summary rather than a fully searchable and listenable transcript.

GPT-4o (BS=0.61): Speaker 3 and Speaker 0 confirm that they have not done anything with removing disfluencies using the NITE object model.

Dialogue:

Speaker 4: Yeah. But I'm still confused cause I thought that's just what Jonathan said that we can't load a massive document of that size.
Speaker 3: It doesn't matter, cause all the calculation's done off-line.
Speaker 4: On the other hand it shouldn't be should be fifty mega-byte in RAM or something, it shouldn't be massive, should it? Actually fifty hundred megabyte is quite big in RAM.
Speaker 0: Hmm.
Speaker 4: Just thinking, what's the so We do get an error message with the project if we load everything into the project with all the data they load. So we know that doesn't work. So our hope is essentially that we load less into it.
Speaker 1: Yes.
Speaker 0: Yeah.
Speaker 4: What's this lazy loading thing, somebody explain lazy loading to me.

Question: Should loading fifty megabytes of audio data into RAM be considered too massive?

Gold: Speaker 4 considers that fifty to a hundred megabytes in RAM is quite large. He highlights that an error occurs when trying to load all data into the project, implying that this amount of data may be too massive for their purposes and they hope to load less into RAM to avoid issues.

LLaMA3.1-8B (BS=0.52): Speaker 4 considers that loading fifty megabytes of audio data into RAM might be too massive.

GPT-3.5 Turbo (BS=0.35): Speaker 4 considers that loading fifty megabytes of audio data into RAM is not too massive. They discuss the possibility of chunking in pre-annotated audio and using a baseline for the parts that are not pre-annotated. They also mention the importance of prioritizing packages and prototype development.

GPT-4o (BS=0.41): Speaker 4 considers that loading fifty megabytes of audio data into RAM might be quite big and potentially massive, given concerns about the total memory complexity. They suggest checking the capabilities of the audio player and how files can be streamed or buffered to handle the data efficiently.

Figure 2: QA examples from MeetingQA2. BS represents BERTScore.