Counterfactual Analysis for Spoken Dialogue Summarization

Tanushree Banerjee[*]
Department of Computer Science
Princeton University
Priceton, NJ 08544
tb21@princeton.edu

Kiyosu Maeda[*]
Department of Computer Science
Princeton University
Priceton, NJ 08544
km9567@princeton.edu

Abstract

Recent advancements in Large Language Models (LLMs) have enabled us to access high-quality summaries of written sentences, which helps us understand contents quickly and easily. However, summarizing spoken dialogues such as podcasts and interviews has been challenging. Unlike written sentences, spoken dialogue summarization requires multiple upstream tasks (e.g., speaker diarization and speech recognition) that are error-prune and might cause errors on downstream tasks. Understanding how upstream task performance affects the quality of LLM-produced summaries will benefit both end-users and developers who want to analyze why LLMs fail to summarize spoken dialogues. We conducted a technical experiment to understand the effect of errors in speaker diarization and speech recognition on LLMs' summarization performance. To handle various patterns of errors, we leverage counterfactual analysis with which we automatically inject speaker diarization or speech recognition errors into spoken dialogues. The experiment results suggest that as errors in speech recognition tasks increase, the summarization quality decreases, according to GPT-4's evaluation and qualitative analysis. The code for our project can be found here.

1 Introduction

Spoken dialogues, such as interviews, podcasts, and meetings, are rich sources of information. People often listen to or watch those dialogues to obtain information about various topics. Despite their benefits, spoken dialogues are challenging to skim or browse, especially when the duration of those audios is long [14]. It is hard to find essential scenes or utterances from those dialogues by manipulating audio or video interfaces, and people might not want to spend long periods listening to the entire audio.

Recent advancements in natural language processing (NLP) have had the potential to create short summaries of long sentences for us to navigate sentences quickly and easily without listening to entire audio. Particularly, sequence-to-sequence models represented by Large Language Models (LLMs) [32] enable the generation of high-quality and factually consistent summaries from written sentences. Some research reported that LLMs already outperformed humans in summarizing scientific articles [21].

Unlike written text summarization, spoken dialogue summarization has several unique challenges. For example, spoken dialogues often contain disfluencies, such as fillers, slips of the tongue, and repetitions, which makes it difficult to find necessary sentences for summarization [29]. Also, while many questions and answers in dialogues usually contain important information, identifying such question-answer pairs is difficult because speakers do not necessarily ask questions or answer

[*] denotes equal contribution.

questions in an organized manner in spoken dialogue [1]. Those challenges hurt downstream summarization quality. Furthermore, spoken dialogue summarization consists of multiple upstream tasks, such as speaker identification (or diarization) and speech recognition. If upstream tasks fail, it will hurt the performance of the downstream summarization task. When a speech recognition model incorrectly recognizes significant utterances or words, for instance, likely, a summary does not contain that information.

While performance in speaker identification and speech recognition seems to be tightly connected to summarization quality, little was known about the effects of those errors on summarization. We aim to investigate how upstream tasks, in this case, speaker diarization and speech recognition, affect downstream task performance, i.e., spoken dialogue summarization. Specifically, speaker identification models sometimes incorrectly recognize speaker names, and speech recognition models sometimes produce sentences with high Word Error Rates. We will explore how summarization quality changes according to the degrees of those errors. Understanding the effect will benefit end-users and developers who want to understand the reason behind poor summarization quality in light of AI explainability.

Since there are too many error patterns for speaker identification and speech recognition, it is challenging to analyze all the patterns. Our method leverages counterfactuals [17] to generate errors automatically. Specifically, we generate two types of errors: for speaker identification, speaker names are randomly changed to other speaker names, while for speech recognition, words in sentences are randomly masked. By those perturbations, we can readily analyze changes in summarization quality due to various degrees of error in upstream tasks.

We experimented with the AMI corpus as a spoken dialogue dataset and GPT-4 as a summarization model. Results of our technical experiments showed that there was a correlation between the accuracy of upstream tasks and the quality of LLM-generated summaries. As errors in speech recognition increased, the quality of the summaries produced by the LLMs correspondingly decreased, while speaker diarization errors did not affect the quality. However, the results also indicated that the GPT-4 was, to some extent, robust to upstream task errors and could complement these errors. We envision that our method helps to understand the failure of systems with complicated structures where upstream and downstream tasks are tightly connected.

2 Related Work

2.1 Written text summarization

Improving automatic text summarization has received a lot of research attention. Numerous studies have proposed myriad methods, such as Integer Linear Programming (ILP) [8], the Page Rank algorithm [9], and Maximal Marginal Relevance (MMR) [28]. Those methods have aimed to create comprehensive and consistent summaries [23]. Parveen et al. [20] proposed a method to summarize scientific articles with coherence and no redundancy by solving Mixed Integer Programming. More recently, sequence-to-sequence models represented by BART [12] have shown promising results for summarization tasks.

Besides these approaches, Large Language Models (LLMs) have the potential to summarize sentences [18]. Recent studies have investigated LLMs' performance of summarizing texts in various domains, such as news [25, 31], books [4], clinical notes [7], and code [2], while further research would be needed to investigate their generalizability. Pu et al. [21] reported that human evaluators preferred LLM-generated summaries to human-generated ones due to various aspects, such as factual consistency. In contrast to existing research, we apply LLMs to spoken dialogue summarization tasks, which have task-specific difficulties compared to written text summarization.

2.2 Spoken dialogue summarization

A summary of a spoken dialogue allows users to figure out the contents quickly without watching a whole recording or reading an entire transcript. Much research has made efforts to generate text summaries of spoken dialogues mainly by leveraging existing practices in written text summarization [19]. Unlike written text summarization, however, spoken dialogue summarization has unique challenges. For example, spoken dialogues contain disfluency, such as filled pauses and repetitions, which hurts the summarization quality [29]. Numerous research studies have been conducted to create a summarization model robust to disfluency [10]. Furthermore, it is also challenging to identify pairs of questions and answers in spoken dialogues [1]. If models for such upstream tasks fail to handle them, they will produce summaries inconsistent with actual contents.

Recent studies have proposed methods to solve these spoken-dialogue-specific difficulties [13]. Li et al. [14] proposed three approaches to improve the summaries' readability, coherence, and adequacy for spoken dialogues that contain disfluency. Chen & Yang [5] presented a data augmentation method to train summarization models for conversation. However, little was known about how those challenges affect summarization performance. In this study, we focus on speaker diarization and speech recognition as upstream tasks and explore the effect of errors in those upstream tasks on spoken dialogue summarization. Understanding the relationship between upstream and downstream tasks will benefit end-users and developers who want to know why summarization fails in light of AI explainability. Since there are too many error patterns, it is hard to access them. We leverage the concept of counterfactual explanations to automatically generate errors related to speaker diarization and speech recognition in spoken dialogues.

2.3 Counterfactual explanations in NLP

In NLP, counterfactual explanations have been used to analyze how differences in input texts affect models' predictions [17]. Polyjuice [27] is a method that allows users to generate various counterfactuals through controlling perturbation types (e.g., delete negation, lexical) and locations trained by fine-tuning GPT-2. Ross et al. [24] presents MICE, which generates contrastive explanations of model predictions by editing input texts. Treviso et al. [26] proposed a method to generate more natural counterfactual examples while regularizing selective rationales.

Inspired by existing research, we proposed a method to inject speaker diarization or speech recognition errors into spoken dialogues to analyze how the model performance changes according to those errors. While existing approaches focus on counterfactuals for a single sentence, our method deals with spoken dialogues that contain hundreds of sentences. More specifically, the method randomly changes the identified speakers of each sentence or masks words in each sentence. In doing so, the speaker diarization and speech recognition error rates will increase.

3 Method

We aim to investigate how upstream tasks (speaker identification and speech recognition) affect summarization task performance. We explore the effect by leveraging counterfactuals, namely, automatically generating errors related to speaker diarization and speech recognition in the spoken dialogue dataset and measuring the quality of summarization generated by LLMs according to the levels of errors we inject. Unlike traditional summarization models, however, LLMs might have the common sense to understand contents containing those errors, as human beings do. Thus, we set the following two hypotheses.

- H1: As errors in speaker diarization and speech recognition increase, summarization quality will decrease.
- H2: LLMs might already have the reasoning ability robust to upstream task errors and produce summaries with less quality degradation even if the errors in speaker diarization and speech recognition increase.

3.1 Data set

We use the AMI corpus [11] as spoken dialogue dataset. This corpus consists of 100 hours of meeting recordings (e.g., videos and audio). It also contains annotated summaries and transcripts with speaker names for each meeting. Among the corpus, we used three meetings for analysis in which participants discussed a technical project. An annotation of each meeting has utterances by multiple speakers. There are four speakers in every meeting. We create a dialogue (string format in Python) as a part of the inputs for the summarization model. The format of the dialogue is described as follows:

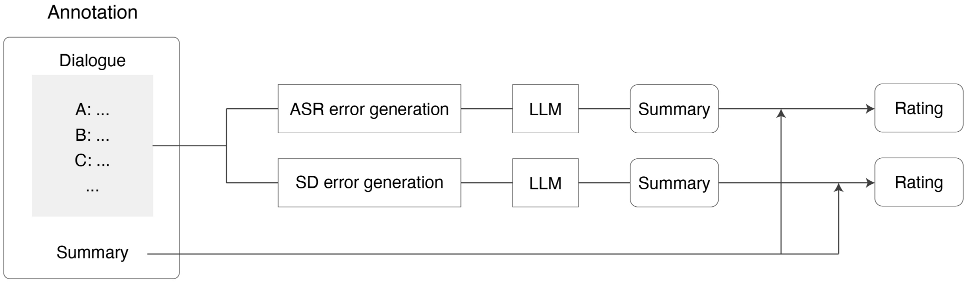

Figure 1: The pipeline of the method. It first injects automatic speech recognition (ASR) or speaker diarization (SD) errors in a dialogue. Then, it creates a summary for the dialogue using a Large language model (LLM). Finally, the pipeline rates the summary by comparing an annotated summary using either a BERT score or an LLM.

$$
\begin{aligned}
&\text{dialogue} = \\
&\text{“speaker\_name1 : utterance1,} \\
&\text{speaker\_name2 : utterance2,} \\
&\text{speaker\_name1 : utterance3,} \\
&\text{speaker\_name3 : utterance4,} \\
&\text{...”}
\end{aligned}
$$

In the original annotation, speaker names (“speaker_name1”, “speaker_name2”, and “speaker_name3” in the above texts) are one-character alphabet (e.g., “A”, “B”, “C”, and “D”). However, specific names are used in sentences in each meeting (e.g., Gabriel and Catherine).

3.2 Upstream tasks and counterfactual error generation

We focus on two upstream tasks for spoken dialogue summarization: speaker diarization and speech recognition. Since the corpus already has annotated speaker names and transcripts, we generate counterfactual errors related to those two tasks. The overview of counterfactual error generation is shown in Figure 2.

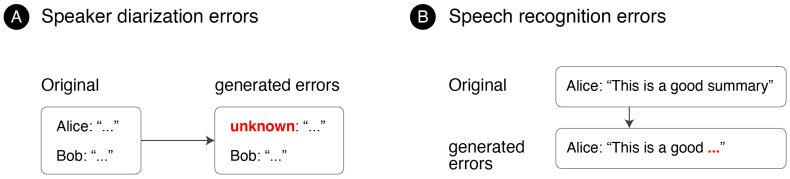
A Speaker diarization errors
Original
generated errors
Alice: “...”
Bob: “...”
unknown: “...”
Bob: “...”
B Speech recognition errors
Original
Alice: “This is a good summary”
generated errors
Alice: “This is a good ...”

Figure 2: (A) Speaker identification errors. In this example, the original speaker’s name is “Alice”, while after error generation, the speaker’s name is changed to “unknown”. (B) Speech recognition errors. In this example, an original sentence is “This is a good summary”, while after error generation, the word “summary” is masked: “This is a good ...”.

3.2.1 Speaker diarization

For speaker diarization (SD), we control the Diarization Error Rate (DER) by randomly changing speaker names to “unknown” as described in Figure 2 A. DER is often used measure to estimate the performance of speaker identification models, which can be calculated in our problem setting as follows:

$$
\text{DER} = \frac{\text{\# missed detection}}{\text{the total number of sentences}} \tag{1}
$$

where "# missed detection" refers to the number of errors we incorporate into one dialogue. This is a simplified formulation from the original one that considers false alarm (the duration of non-speech segments wrongly classified as speech). In our context, however, we only need to consider missed detection since our method does not generate errors related to false alarms. Note that original annotations in the dataset have no diarization error, which means DER $= 0$. We measure how summarization performance changes according to different levels of DER. Specifically, we generate summaries with different DERs $(0, 10, 20, 30, 40)$ for each meeting.

3.2.2 Speech recognition

For speech recognition, we control the Word Error Rate (WER). Specifically, we randomly mask words in annotated sentences as described in Figure 2 B. In our problem setting, WER can be calculated as follows:

$$\text{WER} = \frac{\text{\# substitutions}}{\text{\# words in a sentence}} \tag{2}$$

where "# substitutions" refers to the number of masked words in each sentence. This is also a simplified formulation from the original one, considering the number of deletions and insertions since our proposed method only substitutes words. Similarly, in speaker diarization, original annotations in the dataset have no word error (WER $= 0$), and we measure and analyze how summarization performance changes as WER increases. Similarly to DER, we generate summaries with different WERs $(0, 10, 20, 30, 40)$ for each meeting.

3.3 Summarization & Metrics

We use GPT-4 [1] as a summarization model since it is considered a top-tier large language model for various tasks. We set two types of prompts: error-unaware and error-aware prompts, corresponding to H1 and H2. The error-unaware prompt does not assume that sentences might contain errors related to upstream tasks, while the error-aware prompt does. We show the two prompts below.

1. Error-unaware prompt

1.1 For abstractive summaries

prompt = "Please generate an abstractive summary for the following dialogue: ${dialogue}"

2. Error-aware prompt

2.1 For abstractive summaries

prompt = "Please generate an abstractive summary for the following dialogue.
Note that this dialogue might contain errors in speaker names and transcripts: ${dialogue}"

To measure the quality of generated summaries, we let an LLM rate them with a 5-point scale as metrics. Evaluating text generation performance by LLMs has been studied and adopted in various tasks [6, 33]. AMI corpus has two kinds of annotated summarizations for each meeting: abstractive and extractive [3]. An abstractive summary consists of unique sentences given a context, while an extractive summary is a set of sentences selected from inputs. In this experiment, we used only abstractive ones. As shown below, we have two prompts for generating abstractive summaries. We have two types of summaries in total and compare them with annotated abstractive summaries using LLM's evaluations. We describe how an LLM evaluated summaries below, using a similar prompt proposed in [33].

[1] https://cdn.openai.com/papers/gpt-4.pdf

Prompt for evaluating summaries

prompt = Could you please grade a summary for the following dialogue? you must rate the response on a scale of 1 (low) to 5 (high).

[The start of the dialogue]
${annotated dialogue}
[The end of the dialogue]

[The start of the summary]
${generated summary}
[The end of the summary]

We also evaluated some summaries qualitatively by reading through summaries, especially when scores by GPT-4 were low. In this case, we checked not only the summaries themselves but also the outputs of GPT-4.

4 Results

Figure 3 and Figure 4 show the experiment results. We analyzed the relationship between DER or WER and summarization performance.

4.1 DER and summarization performance

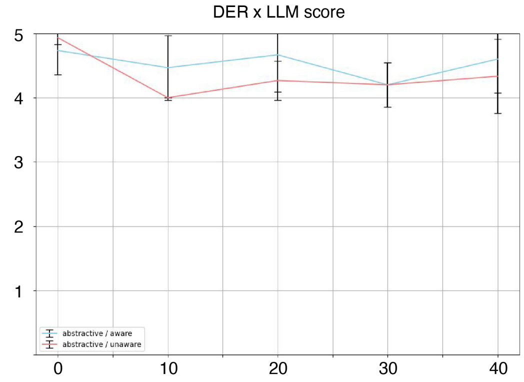

Figure 3: DER and scores evaluated by LLM: Each plot shows the results on error-aware and error-unaware prompts. A red line shows the results for error-unaware prompts, while a blue line shows the results for error-aware prompts.

Figure 3 shows scores evaluated by GPT-4. Overall, summaries were highly rated by GPT-4 regardless of the degrees of speaker diarization errors. Also, we could not observe the apparent difference in summaries between error-aware (a blue line) and error-unaware (a red line) prompts. These results indicated that speaker diarization errors generated by our proposed method did not affect summarization quality much. As the number of unknown speakers increased. However, we found a tendency for speaker names to appear less in the summaries. For instance, in one of the meetings, specific speaker names such as Gabriel and Reissa were referred to when DER = 0. On the other hand, there was no name in a summary when DER = 40, which means that each sentence was not connected to a participant in the meeting. While the LLM's evaluation did not show any difference, this might hurt user experiences when users want to check their subsequent actions after the meeting.

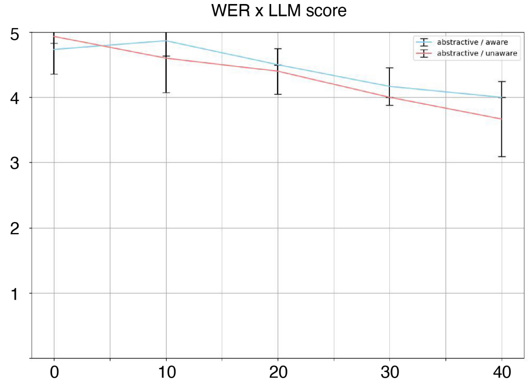

Figure 4: WER and scores evaluated by LLM. Each plot has results for both error-aware and error-unaware prompts. A red line shows the results for error-unaware prompts, while a blue line shows the results for error-aware prompts.

4.2 WER and summarization performance

Figure 4 shows scores evaluated by GPT-4. Unlike the results of DER and summarization performance, we could observe as WER increased, scores rated by GPT-4 decreased. Also, there was little difference in scores between error-aware and error-unaware prompts. When WER was high, generated summaries became more ambiguous, lacked detailed descriptions of the meetings, and were sometimes even wrong, which made it difficult for participants to catch up with the meeting only with summaries. According to GPT-4's rationale on grading three out of five for one of the generated summaries (WER $= 0$), for instance, a generated summary could not figure out the contents of the meeting and mentioned unrelated things. Furthermore, it could not capture the meeting's specific focus and technical details. Since that information is essential in the project meeting, lacking technical details might affect the summary's quality estimated by GPT-4. We added one example of rating by GPT-4 in Appendix A.1.

4.3 Limitations

While we found some differences in summarization quality depending on DER or WER, our experiments have limitations.

We used GPT-4 for rating the summaries produced by GPT-4. This method has been increasingly investigated and utilized in recent research (e.g., [33]). However, there are other methods adopted for evaluating generated texts. For example, ROUGE [15], BLUE [16], and BERT score [30] are the most well-known approaches to measure summarization quality automatically. Also, human-generated rating is still an important metric and usually outperforms automatic evaluation approaches, though it takes time. It is possible that LLMs still cannot figure out the details of generated summaries, which might affect the evaluation quality. In the future, it will be possible to evaluate summaries with those methods or compare scores rated by each method.

Furthermore, we only used a single-domain dataset (project meetings) for evaluation. As described in Section 1, there are myriad domains, such as interviews. Further research would be needed to investigate how our results are generalizable. Note that it has been challenging to collect spoken dialogue datasets, and there are few available datasets for spoken dialogues [14], which makes it difficult to evaluate in diverse application areas.

5 Discussion

5.1 Are LLMs robust to upstream task errors?

As in our experiments' results, DER did not affect summarization quality, while as the WER in transcripts increased, summarization quality decreased. The results partly support H1. However, the LLM could still produce good summaries, considering that average scores were above 4.0 when WER = $10, 20$, or 30. This indicated that LLMs might be somewhat robust to speech recognition errors regardless of prompt types (error-aware or error-unaware), which partly supports H2. Nevertheless, further research will be needed to extensively investigate the robustness of the upstream task errors, like comparing GPT-4 with other LLMs (e.g., Xwin [2], mpt [3], vicuna [4], Llama-2 [5]). Since our proposed method has a modular architecture, it can be easily extended to those models.

5.2 Towards more realistic and powerful counterfactual error generation

While we took a simple approach to generate errors in speaker names and transcripts so we could control error rates easily, there would be different methods to create errors that appear in real-world scenarios. For example, there were speaker names in annotated transcripts, which was one of the reasons why the LLM was not affected by diarization error injection. Investigating how LLMs' performance changes when we change speaker names in sentences will be interesting. Another approach is to use automatic speech recognition (ASR) models. While the current ASR models, such as whisper [22], can produce authentic transcripts with low WER, it is still challenging to transcribe noisy audio correctly. It is also challenging for the models to transcribe proper nouns or technical terms that often contain important information. Thus, we can naturally generate transcripts with higher word error rates by adding noise to audio and replacing original sentences with generated ones. In those cases, we must consider the number of deletions and insertions besides substitutions, as seen in Equation 2. While those approaches can produce sentences that contain more realistic errors, noting that it would be hard to control error rates with them.

6 Conclusion

We investigated how upstream tasks, speaker diarization, and speech recognition affect downstream summarization task performance. Our proposed pipeline automatically generated counterfactual errors in dialogue transcripts related to speech recognition (WER) and speaker diarization (DER), changing the degrees of errors. Then, we generated summaries and evaluated them using GPT-4, a large language model. We could observe differences in the performance of summaries depending on DER and WER. Speaker names that did not appear in the summaries or the details of the meetings were not included in the summaries. We envision that those findings will benefit end-users and developers who try to understand how errors caused by upstream tasks affect summarization performance or why summarization fails.

References

[1] Stergos D. Afantenos, Eric Kow, Nicholas Asher, and Jérémy Perret. Discourse parsing for multi-party chat dialogues. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 928–937. The Association for Computational Linguistics, 2015.

[2] Toufique Ahmed and Premkumar T. Devanbu. Few-shot training llms for project-specific code-summarization. In *37th IEEE/ACM International Conference on Automated Software Engineering, ASE 2022, Rochester, MI, USA, October 10-14, 2022*, pages 177:1–177:5. ACM, 2022.

[3] Giuseppe Carenini and Jackie Chi Kit Cheung. Extractive vs. nlg-based abstractive summarization of evaluative text: The effect of corpus controversiality. In *INLG 2008 - Proceedings of the Fifth International Natural Language Generation Conference, June 12-14, 2008, Salt Fork, Ohio, USA*. The Association for Computer Linguistics, 2008.

[2] https://huggingface.co/Xwin-LM/Xwin-LM-7B-V0.1

[3] https://huggingface.co/mosaicml/mpt-7b-chat

[4] https://huggingface.co/lmsys/vicuna-7b-v1.3

[5] https://huggingface.co/meta-llama/Llama-2-7b-hf

[4] Yapei Chang, Kyle Lo, Tanya Goyal, and Mohit Iyyer. Booookscore: A systematic exploration of book-length summarization in the era of llms. *CoRR*, abs/2310.00785, 2023.

[5] Jiaao Chen and Diyi Yang. Simple conversational data augmentation for semi-supervised abstractive dialogue summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 6605–6616. Association for Computational Linguistics, 2021.

[6] David Cheng-Han Chiang and Hung-yi Lee. Can large language models be an alternative to human evaluations? In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 15607–15631. Association for Computational Linguistics, 2023.

[7] Yu-Neng Chuang, Ruixiang Tang, Xiaoqian Jiang, and Xia Hu. Spec: A soft prompt-based calibration on mitigating performance variability in clinical notes summarization. *CoRR*, abs/2303.13035, 2023.

[8] Greg Durrett, Taylor Berg-Kirkpatrick, and Dan Klein. Learning-based single-document summarization with compression and anaphoricity constraints. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics, 2016.

[9] Günes Erkan and Dragomir R. Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Intell. Res.*, 22:457–479, 2004.

[10] Mark Johnson and Eugene Charniak. A tag-based noisy-channel model of speech repairs. In Donia Scott, Walter Daelemans, and Marilyn A. Walker, editors, *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics, 21-26 July, 2004, Barcelona, Spain*, pages 33–39. ACL, 2004.

[11] Wessel Kraaij, Thomas Hain, Mike Lincoln, and Wilfried Post. The ami meeting corpus. In *Proc. International Conference on Methods and Techniques in Behavioral Research*, 2005.

[12] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. Association for Computational Linguistics, 2020.

[13] Daniel Li, Thomas Chen, Albert Tung, and Lydia B. Chilton. Hierarchical summarization for longform spoken dialog. In *UIST '21: The 34th Annual ACM Symposium on User Interface Software and Technology, Virtual Event, USA, October 10-14, 2021*, pages 582–597, New York, NY, 2021. ACM.

[14] Daniel Li, Thomas Chen, Alec Zadikian, Albert Tung, and Lydia B. Chilton. Improving automatic summarization for browsing longform spoken dialog. In Albrecht Schmidt, Kaisa Väänänen, Tesh Goyal, Per Ola Kristensson, Anicia Peters, Stefanie Mueller, Julie R. Williamson, and Max L. Wilson, editors, *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems, CHI 2023, Hamburg, Germany, April 23-28, 2023*, pages 106:1–106:20, New York, NY, 2023. ACM.

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

[16] Chin-Yew Lin and Eduard H. Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, HLT-NAACL 2003, Edmonton, Canada, May 27 - June 1, 2003*. The Association for Computational Linguistics, 2003.

[17] Andreas Madsen, Siva Reddy, and Sarath Chandar. Post-hoc interpretability for neural NLP: A survey. *ACM Comput. Surv.*, 55(8):155:1–155:42, 2023.

[18] Bonan Min, Hayley Ross, Elior Sulem, Amir Pouran Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heintz, and Dan Roth. Recent advances in natural language processing via large pre-trained language models: A survey. *ACM Comput. Surv.*, 56(2):30:1–30:40, 2024.

[19] Ani Nenkova and Kathleen R. McKeown. Automatic summarization. *Found. Trends Inf. Retr.*, 5(2-3):103–233, 2011.

[20] Daraksha Parveen, Mohsen Mesgar, and Michael Strube. Generating coherent summaries of scientific articles using coherence patterns. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 772–783. The Association for Computational Linguistics, 2016.

[21] Xiao Pu, Mingqi Gao, and Xiaojun Wan. Summarization is (almost) dead. *CoRR*, abs/2309.09558, 2023.

[22] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 28492–28518. PMLR, 2023.

[23] Pengjie Ren, Furu Wei, Zhumin Chen, Jun Ma, and Ming Zhou. A redundancy-aware sentence regression framework for extractive summarization. In *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, pages 33–43. ACL, 2016.

[24] Alexis Ross, Ana Marasovic, and Matthew E. Peters. Explaining NLP models via minimal contrastive editing (mice). In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 3840–3852. Association for Computational Linguistics, 2021.

[25] Derek Tam, Anisha Mascarenhas, Shiyue Zhang, Sarah Kwan, Mohit Bansal, and Colin Raffel. Evaluating the factual consistency of large language models through news summarization. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 5220–5255. Association for Computational Linguistics, 2023.

[26] Marcos V. Treviso, Alexis Ross, Nuno Miguel Guerreiro, and André Martins. CREST: A joint framework for rationalization and counterfactual text generation. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 15109–15126. Association for Computational Linguistics, 2023.

[27] Tongshuang Wu, Marco Túlio Ribeiro, Jeffrey Heer, and Daniel S. Weld. Polyjuice: Generating counterfactuals for explaining, evaluating, and improving models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 6707–6723. Association for Computational Linguistics, 2021.

[28] Yong-Dong Xu, Xiao-Dong Zhang, Guang-Ri Quan, and Ya-Dong Wang. MRS for multi-document summarization by sentence extraction. *Telecommun. Syst.*, 53(1):91–98, 2013.

[29] Klaus Zechner. Summarization of spoken language-challenges, methods, and prospects. *Speech technology expert eZine*, 6, 2002.

[30] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[31] Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen R. McKeown, and Tatsunori B. Hashimoto. Benchmarking large language models for news summarization. *CoRR*, abs/2301.13848, 2023.

[32] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. A survey of large language models. *CoRR*, abs/2303.18223, 2023.

[33] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. *CoRR*, abs/2306.05685, 2023.

A Appendix

A.1 GPT-4's output example when rating a generated summary

The sentences below were GPT-4's rating results on one of the generated summaries.

The summary provided for the dialogue is quite detailed and captures many elements of the conversation, but it also includes some inaccuracies and misses key aspects of the dialogue. Based on this, I would rate it a 3 out of 5. Here's a breakdown of the assessment:

Relevance and Accuracy (2/5): The summary does not accurately reflect the content of the dialogue. It mentions a team-building exercise involving favorite animals, which is not present in the dialogue. Additionally, the dialogue is more focused on the technical and practical aspects of designing a remote control, such as energy sources, signal types, and ergonomic considerations, rather than a broad design process.

Completeness (3/5): The summary captures some aspects of the dialogue, such as the discussion about the remote control design and team roles. However, it fails to mention significant parts of the conversation, like the discussion on batteries, infra-red vs. Bluetooth, ergonomics, and specific user feedback on remote controls. These elements are crucial to understanding the team's challenges and objectives.

Clarity and Structure (4/5): The summary is well-structured and presents information in a logical order. The introduction, main body, and conclusion are clearly defined, making it easy to follow.

Conciseness (3/5): While the summary is somewhat concise, the inclusion of irrelevant information (such as the animal exercise) detracts from its conciseness. It could be more focused on the key topics discussed in the meeting.

Overall Understanding (3/5): The summary shows a general understanding of a team meeting about a project, but it does not fully capture the specific focus and technical details of the dialogue. The emphasis on user-friendliness and market strategy is aligned with the dialogue, but the details are not completely accurate or complete.

In summary, the summary is decent but could be improved by removing the inaccuracies and including more specific details from the dialogue, particularly those related to the technical discussions and design challenges of the remote control project.