



An Exploratory Framework for LLM-assisted Human Annotation of Speech Datasets

Alexander Johnson¹, Harsh Deshpande¹, Emmy Phung¹, Ahmad Emami¹

¹Machine Learning Center of Excellence, JPMorganChase, USA

alexander.aaa.johnson@jpmorganchase.com

Abstract

We introduce a framework for LLM-based human-in-the-loop ASR designed to enhance the quality of ASR transcripts, with a particular focus on accurately capturing named entities. A key contribution of this work is demonstrating that when LLMs are provided with high-quality, human-annotated transcript examples, even a small set can significantly improve WER and entity recall rates. Our framework drastically reduces the costly need for human annotation to just 5% of the entire call. The system outperforms all baselines, including out-of-the-box Whisper and Whisper with a zero-shot GPT corrector. In this work, we derive insights on how a chain-of-thought framework can effectively utilize LLM prompts and human input to improve speech data annotation quality. We achieve a 10% or greater relative improvement in WER and entity F1 score over the baseline with a minimal amount of human effort. **Index Terms:** speech data annotation, human-in-the-loop, speech recognition, large language models

1. Introduction

It is often time-consuming and expensive for human annotators to label new large speech datasets for training automatic speech recognition (ASR) and spoken language understanding (SLU) systems. A key research question then becomes how we can most efficiently use annotator resources to label new speech data. Currently, many speech transcript annotators will start by producing an ASR transcript and then carefully scan over it to correct any ASR errors. Given this process, our research question could also be phrased as: How much an ASR transcript do annotators actually need to spend time looking at to ensure high enough data quality? Many ASR systems, including the widely-used Whisper model [1], have achieved state-of-the-art performance by training on datasets that are at least partially pseudo-labeled by other machines [2, 3, 4]. This might suggest that large portions of speech datasets can be effectively labeled by ASR systems, and human annotators are better reserved for the portions that are difficult for speech recognition to handle. For example, many ASR systems perform well for common English words but struggle with out-of-vocabulary (OOV) words (ie. words that were not seen during training) such as lesser known named entities (eg. people’s names, location names, and company names) and novel terms (eg. the name of a recently created group or event) [5]. In this case, it may be advantageous to use ASR to transcribe the majority of the speech data and then ask human annotators only to correct out-of-vocabulary words or other segments for which the ASR performs poorly. Large language models (LLMs) offer an attractive solution to this problem. By parsing through ASR transcripts, an LLM can identify many of the areas in which an ASR system may be

known to produce higher word error rates (OOV words, easily confused homophones, etc.). LLMs have been shown to be effective in correcting errors in ASR transcripts (eg. [6, 7, 8]). Notably, the work in [9] uses a text-only approach to train an LLM to predict words with a high probability of being transcribed with ASR errors. The authors of [10] tackle the performance issue that many ASR models exhibit with OOV entities by training a model with the Whisper encoder and an LLM decoder to jointly improve ASR and named entity recognition (NER) performance. While using language models to improve ASR outputs has been a common practice for years [11], the capabilities of LLMs in this task has yet to be fully explored.

However, correcting ASR transcripts with LLMs includes several challenges. First, the LLM may be overzealous in improving the syntax and spelling of the input text at the cost of making the transcript less accurate to what was actually said [12]. Second, the LLM may not have knowledge of some words that were said in a transcript, such as domain-specific terminology or entities, and will simply map them to tokens that are more likely to appear in written language. Therefore, we instead propose leveraging the capabilities of an LLM to reduce the workload of a human annotator as opposed to a fully automated approach. Our proposed workflow is to 1) Use an LLM to retrieve segments from an ASR transcript that are more likely to require the attention of a human annotator, 2) Obtain the human transcription for those segments only, and 3) again use the LLM to correct the errors present in the selected segments across the ASR transcript. Human-in-the-loop LLM-based strategies have proven effective for text-generation [13], entity classification [14], and other tasks. In order to adapt such techniques for speech, we first investigate zero shot and few shot strategies [15] for prompting an LLM to retrieve the segments from an ASR transcript that would benefit most from human intervention. Given human-written transcripts for these segments, we then investigate strategies to use an LLM to best extrapolate these corrections to other portions of the transcript without over-correcting words that have been transcribed accurately. Novel contributions include 1) the creation of a framework for LLM-based human-in-the-loop ASR transcript 2) a strategy for the effective selection of transcript or audio segments for human review and 3) an improvement in performance in both word error rate and entity F1 score over using an LLM alone.

2. Methods

To reduce the amount of human intervention needed to produce human-annotated transcripts from ASR transcripts, we employ the framework shown in Figure 1. First, we generate an ASR transcript for a given audio file. Then we ask an LLM to re-

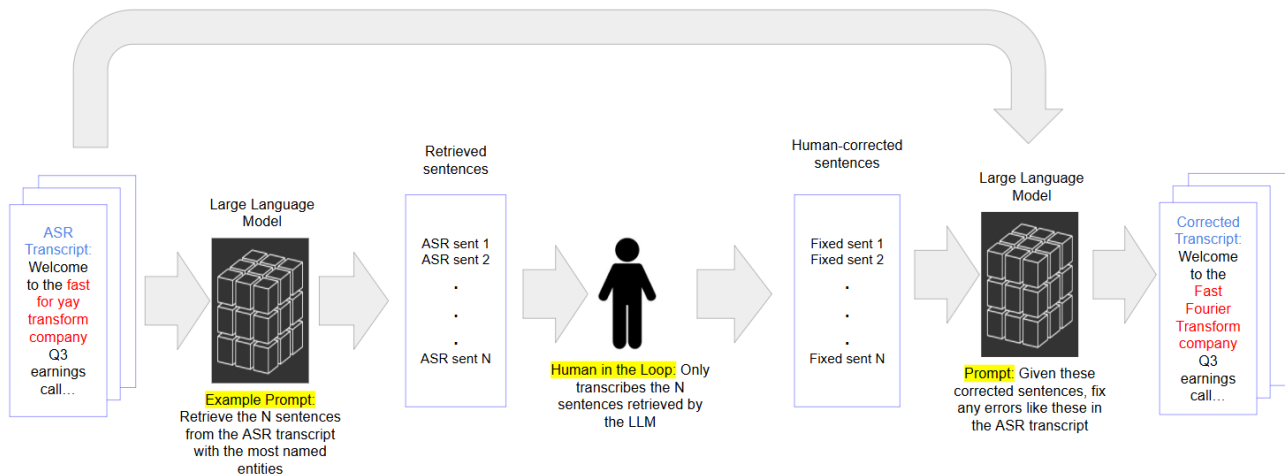


Figure 1: Pipeline for the proposed framework: An ASR transcript is passed to an LLM that is prompted to retrieve N sentences from the transcript that are representative of the ASR transcripts throughout. Then a human is asked to correct only those N sentences. Last, the LLM given the human input and is asked to make those corrections across the entire transcript.

retrieve a set of sentences from the ASR transcript based on a given criterion. The goal of this step is to extract sentences that most need a human annotator to correct them or are representative of overarching errors in the ASR transcript. We experiment with prompt engineering to identify the most effective prompts to achieve this task. A human is then tasked with manually correcting only this subset of retrieved sentences from the ASR transcript. Next, the LLM is given the corrected sentences as few shot examples along with the ASR transcript, and the LLM is asked to make the corrections seen in the annotator’s work across the entire transcript.

2.1. Dataset

In our experiments, we use the Earnings 21 dataset [16]. This English-language dataset consists of 44 recordings of corporate earnings calls totaling approximately 39 hours of speech. We choose this dataset because it is rich with labeled entities (company names, names of company employees, regulatory bodies, etc.) that can be used for NER, making it a good test case to see how well our framework improves recall of out-of-vocabulary entities in transcripts. The entities in the Earnings 21 dataset are labeled by category (PERSON, ORDINAL NUMBER, ORGANIZATION, DATE, etc.). The authors of [16] report that many of these categories, such as DATE, TIME, and ORDINAL numbers are trivial for ASR systems to transcribe correctly. As such, we only focus on the three entity categories reported in [16] as the most difficult to correctly transcribe: PERSON (People, including fictional), ORG (Companies, agencies, institutions, etc.), and FAC (Buildings, airports, highways, bridges, etc.). We split the dataset randomly into 40% validation and 60% test data. The validation set is used for tuning hyper-parameters while the results are reported on the test set.

2.2. Experiments

We first transcribe each recording with Whisper Medium-En [1], a state-of-the-art ASR system. Next, we seek to use the LLM to retrieve the N sentences whose human annotation would best help the language model correct the entire ASR transcript. We try a variety of prompts and report the best here:

- List the sentences from the following call transcript that contain the most errors or do not make logical or grammatical sense.
- List the sentences from the following call transcript that could best be used to summarize the call transcript.
- List the sentences from the following call transcript that are most unique or unlikely to be found in another call transcript
- List the sentences from the following call transcript that contain the highest number of named entities.
- List the sentences from the following call transcript that contain the highest number of named entities that are likely misspelled or contain an error.

In addition to these instructions, the LLM is 1) asked not to produce any additional text or text not found directly in the call transcript, and 2) given three fictitious examples of sentences that fit the given criterion for few shot prompting. We perform our validation (prompt and parameter tuning) experiments in this task using GPT-4o [17].

Given N sentences from the ASR transcript, we then align the ASR transcript to the ground truth transcripts in the Earnings 21 dataset and retrieve the corresponding human annotation for that ASR transcript segment. The LLM is then asked to correct the ASR transcript given the selected ground truth sentences as a template. Many LLMs, including the models used here, have a maximum input and output token length that is smaller than the length of a longer call transcript. To circumvent this, each ASR transcript is broken into chunks of 10 sentences, and the LLM is called to correct one chunk of the ASR transcript at a time. The output corrected transcript chunks are then concatenated to form the entire corrected transcript. We notice that the LLM is occasionally prone to hallucinating (adding extraneous or erroneous text into the output transcript) or may refuse to complete the task based on the content of a given transcript chunk. Therefore, we additionally experiment with a word error rate (WER)-based threshold for rejecting changes to the target transcript chunk. That is, if the WER between the ASR transcript chunk and the LLM-corrected transcript chunk, using the ASR transcript as the reference, is greater than a given threshold, we reject the LLM output and instead use the original ASR

transcript chunk in the final transcript output. We find through experimentation on the validation set that a WER threshold of 25% works well for this.

After identifying the prompts and hyper-parameters that produce the best performance with GPT-4o, we attempt to replicate these metrics with Llama3-8b another large language model of a comparable that has open-source implementation. We find through initial testing that the out-of-box Llama3-8b does not have the capacity to follow an instruction for a task such as “correct the following ASR transcript” well. Therefore, we fine-tune the Llama3-8b model using QLoRA [18]. The training examples are segments of the ASR transcripts of the validation set chunked into 10 sentence segments paired with their corresponding ground truth segment. Llama is then fine-tuned to correct the ASR transcript given the first 10 sentences of the ground transcript as a few-shot example. Fine-tuning is run for one epoch with an AdamW optimizer using a weight decay and learning rate of 1e-3 and 2e-4 respectively. Model weights are quantized to float16 so that the model can be run effectively on a single NVidia Tesla V100 GPU.

3. Results and Discussion

3.1. Metrics

To evaluate the quality of outputs produced by the framework (the entire ASR transcript after few-shot LLM correction), we compute the Word Error Rate (WER), the Entity Recall Rate (%Ent Rec.), the Entity Precision Rate (%Ent. Prec.), and the Entity F1 Score (%Ent F1) as defined in [19]. A lower WER indicates the framework being able to correct spelling mistakes or filling in missing words, whereas a higher entity recall and precision indicates the framework correcting or adding correct entities in the ASR transcript.

3.2. Results and Analysis

Results on our baseline frameworks (shown in Table 1) show a WER of 11.05% and an Entity F1 of 87.50% on the raw whisper transcripts. Naive baselines such prompting the LLM to fix all errors in the ASR transcript without further specificity in the directions and prompting the LLM to fix only the entity errors in the ASR transcript often lead to a degradation in performance with respect to the WER. These strategies do help in increasing the recall rate for entities, but hamper entity precision rate, hinting at ‘over-correction’ from these baselines, where either words or entities that do not need correction are wrongly substituted, or entities are added at incorrect places.

Table 2 shows the results for all prompts described in Section 2.2. Prompts focusing on extracting sentences with error-prone words (“Sentences with the highest number of entities that are prone to being misspelled”) or with more entities (“Sentences with the highest number of entities”) tend to perform better, especially when instances of hallucination are ignored by rejecting LLM outputs that change the ASR transcript by more than the allowed threshold WER. This implies prompts like these are effective at extracting the most impactful candidate sentences to be transcribed manually and fed to the LLM for few-shot error correction. In contrast, prompts that focus on the semantic content of the call, like parts useful for a summary, or most unique parts, perform slightly worse. This may suggest that these prompts do not focus the LLM on error-dense portions of the transcript, and less subjective retrieval criteria is necessary for the LLM to perform consistently.

All results in Table 2 use 10 few shot examples in the LLM

Table 1: *Baseline results. We show the percent word error rate (%WER) and recall, precision, and F1 score (%Ent Rec., %Ent Prec., and %Ent F1 respectively) in capturing entities for each system. We calculate the metrics for the system outputs from 1) the out-of-box Whisper ASR transcripts, the modified ASR transcripts after prompting GPT to correct all errors in them, and the modified ASR transcripts after asking GPT to correct only the entities in them.*

Baseline	%WER↓	%Ent Rec.↑	%Ent Prec.↑	%Ent F1↑
Whisper OOB	11.05	83.72	91.64	87.50
GPT-Fix All Errors	11.80	86.15	90.16	88.11
GPT-Fix Only Ent.	11.80	84.50	87.93	86.18

prompt to correct the ASR prompt. In Table 3, we evaluate the effect of using a different number of few-shot example sentences from the ground truth sentences in the LLM prompt, varying the number of examples from 5 to 20. We perform this experiment with the three best-performing prompts from Table 2: “Sentences that likely have an error” with WER thresholding, “Sentences with the highest number of entities” with no WER thresholding, and “Sentences with the highest number of entities that are prone to being misspelled” with WER thresholding. The strategy of using sentences with likely errors consistently demonstrates robust performance across different sentence counts, maintaining a low Word Error Rate (WER) around 10.75% and achieving an Entity F1 score close to 87.65% to 89.68%. This indicates that the LLM is relatively robust to the number of examples used in prompting, likely due to its focus on addressing specific errors in the ASR output. Therefore, good performance may be achieved with a small number of annotated sentences. We note that prompt “Most entity-rich sentences” called with 5 few-shot examples resulted in hallucination, causing a low-performing WER and entity recall rate (181.71% and 49.72%). This issue is transient for this case, as it does not appear when using 10 or 20 sentences in the few-shot prompt, and is eliminated in the other cases by applying the WER thresholding.

The strategy of using error-prone entity sentences remains stable across different sentence counts, with a WER around 10.89% to 10.91% and an Entity F1 score consistently above 92.39%, highlighting its reliability in improving transcript accuracy by focusing on specific, error-prone entities. Analysis on the sentences retrieved by the LLM for few-shot error correction shows that several of the sentences contain redundant mentions of the same entity when the LLM is asked to retrieve larger numbers of sentences, and so a possible area of improvement is diversifying the entities sentences returned.

Re-running the 10 sentence “Error-prone entity sentences” case using WER thresholding (ie. the case that produced the highest Entity F1 score with 10 few shot examples) with the fine-tuned Llama3-8b model resulted in a **WER of 11.21% and Entity (Recall, Precision, F1) scores of (84.59%, 92.25%, 88.25%)**. While not performing as well as the GPT model, the same parameters and prompt did lead to an improvement over the baseline entity precision-recall metrics with Llama3-8b.

Table 2: Results of using different prompting strategies to have the LLM return 10 sentences of the ASR transcript that match the prompt criterion, retrieving the ground truth version of those sentences, and then using those sentences in few shot prompting to have the LLM revise the ASR transcript

Prompt for Few-Shot Ex. Extraction	Always accept LLM output				Accept LLM output if WER(ASR,LLM_output) <25%			
	%WER↓	%Ent Rec.↑	%Ent Prec.↑	%Ent F1↑	%WER↓	%Ent Rec.↑	%Ent Prec.↑	%Ent F1↑
Sentences that likely have an error	22.82	85.51	83.03	84.25	10.71	86.01	93.67	89.68
Most unique sentences	19.49	85.53	84.91	85.22	10.76	85.39	93.50	89.26
Sentences that form a summary of the transcript	71.91	84.90	57.37	68.48	10.76	85.36	93.47	89.23
Sentences with highest number of entities	10.96	85.95	92.55	89.13	12.55	84.38	90.60	87.37
Sentences with the highest number of entities that are prone to being misspelled	10.87	84.85	91.86	88.22	10.92	91.06	93.76	92.39

Table 3: Results of varying the amount of sentences to be manually annotated using different prompting strategies

Prompt for Few-Shot Ex. Extraction	5 sentences				10 sentences				20 sentences			
	%WER↓	%Ent Rec.↑	%Ent Prec.↑	%Ent F1↑	%WER↓	%Ent Rec.↑	%Ent Prec.↑	%Ent F1	%WER↓	%Ent Rec.↑	%Ent Prec.↑	%Ent F1↑
Sentences with likely errors	10.77	83.31	92.61	87.72	10.72	86.01	93.68	89.68	10.75	83.25	92.55	87.65
Most entity-rich sentences	181.71	91.43	34.14	49.72	10.95	91.15	93.85	92.48	10.85	85.96	92.55	89.13
Error-prone entity sentences	10.89	91.08	93.79	92.42	10.91	91.06	93.76	92.39	10.91	91.08	93.78	92.41

3.3. Discussion

Our proposed LLM-based human-in-the-loop workflow demonstrates significant improvements over both Whisper OOB and LLM-based ASR correction baselines. When tasked with correcting an ASR transcript without references, GPT can rectify some misspelled entities, enhancing the entity recall rate at the expense of a noticeable decrease in precision. The entities that remain uncorrected are usually in the PERSON and ORG categories, often consist of non-English names, and can be easily misspelled if they are unfamiliar. GPT models can correct all instances of the same entity within a transcript, leading to a significant improvements in both recall and precision rates. This supports our hypothesis that GPT models lack knowledge of specific people and organization company names, and greatly benefit from in-context learning. Examples of error-prone and unfamiliar entities include names like "Mueller" and "Collazo" (PERSON) and "Affimed" (ORG), which are mis-transcribed by the baselines as "Miller", "Colorado", and "AFIMET"/"APIMEDS", but are corrected by our workflow.

Despite these advancements, we acknowledge certain limitations in our approach. If a non-English name or error-prone entity is not included in the selected examples, LLMs are unlikely to correct them. One might suggest extracting all entities from a pre-correction ASR transcript and providing a list of candidate entities for human review. However, this approach is susceptible to errors because the original transcript is often noisy and lacks the context needed for annotators to produce high-quality annotations. Annotators require contextual information to distinguish between entities that may sound similar. We apply this same reasoning to LLM few-shot examples, believing that the context provided by these sentences, along with the entities they contain, serves as valuable cues to enhance ASR correction quality. However, knowing that the entities presented in the examples are crucial, it raises the question of how to extract a good segment of the call for human annotation. Our cur-

rent best-performing prompt identifies sentences with the highest number of error-prone entities but does not yet consider the diversity of entities or entity types. A promising direction is to refine LLM prompts to first segment the document and select sentences with the highest counts of error-prone entities from different segments to enhance entity coverage. Another category of errors involves financial abbreviations that are not named entities, which are not yet included in the few-shot examples for LLM and are often mis-transcribed. Although it is not the primary focus of our work, we believe that prompting the LLM to identify not only unique and error-prone entities but also unfamiliar financial acronyms could lead to further improvements in WER.

4. Conclusions

This paper proposes a novel LLM-based pipeline for assisting human annotation of ASR transcripts. As out-of-vocabulary entity words present a large challenge to many ASR systems, our framework uses LLMs to retrieve problem areas of an ASR transcript, gather human input on those sentences, and then attempt to correct them. We share insights on the benefits and shortcomings of different LLM prompting strategies for the given task and improve the entity F1 score of ASR transcripts with both the proprietary GPT-4o model and with the open-source Llama3-8b model. Future steps include training the LLMs with strategies such as reinforcement learning from human feedback to identify which sentences human annotators generally believe to have the most impact on human understanding. Future directions may also involve developing methods for detecting and removing LLM "over-correction" in the ASR transcript at the word-level instead of over several sentences as we perform in this paper so that the LLM output can be considered and implemented at a more granular level.

5. References

- [1] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, “Robust speech recognition via large-scale weak supervision,” in *Proceedings of the 40th International Conference on Machine Learning*, ser. ICML’23. JMLR.org, 2023.
- [2] D. Hwang, K. C. Sim, Z. Huo, and T. Strohmaier, “Pseudo label is better than human label,” *Proc. Interspeech 2022*, pp. 1421–1425, 2022.
- [3] P. Ma, A. Haliassos, A. Fernandez-Lopez, H. Chen, S. Petridis, and M. Pantic, “Auto-avsr: Audio-visual speech recognition with automatic labels,” in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.
- [4] T. Likhomanenko, R. Collobert, N. Jaitly, and S. Bengio, “Continuous soft pseudo-labeling in asr,” in *NeurIPS Workshop*, 2022. [Online]. Available: <https://arxiv.org/abs/2211.06007>
- [5] M. A. B. Jannet, O. Galibert, M. Adda-Decker, and S. Rosset, “How to evaluate asr output for named entity recognition?” in *Interspeech*, 2015, pp. 1289–1293.
- [6] S. Li, C. Chen, C. Y. Kwok, C. Chu, E. S. Chng, and H. Kawai, “Investigating asr error correction with large language model and multilingual 1-best hypotheses,” in *Interspeech 2024*, 2024, pp. 1315–1319.
- [7] Z. Tang, D. Wang, S. Huang, and S. Shang, “Pinyin regularization in error correction for chinese speech recognition with large language models,” in *Interspeech 2024*, 2024, pp. 1910–1914.
- [8] B. Koilakuntla, P. Rana, P. Ahuja, S. Konjeti, and J. Vepa, “Leveraging large language models for post-transcription correction in contact centers,” in *Interspeech 2024*, 2024, pp. 2038–2039.
- [9] Y. Ma, Z. Liu, and O. Kalinli, “Correction focused language model training for speech recognition,” in *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 10 856–10 860.
- [10] Y. Li, J. Yu, M. Zhang, M. Ren, Y. Zhao, X. Zhao, S. Tao, J. Su, and H. Yang, “Using large language model for end-to-end chinese asr and ner,” in *Interspeech 2024*, 2024, pp. 822–826.
- [11] K. Heafield, “KenLM: Faster and smaller language model queries,” in *Proceedings of the Sixth Workshop on Statistical Machine Translation*, C. Callison-Burch, P. Koehn, C. Monz, and O. F. Zaidan, Eds. Edinburgh, Scotland: Association for Computational Linguistics, Jul. 2011, pp. 187–197. [Online]. Available: <https://aclanthology.org/W11-2123/>
- [12] Y. Li, X. Wang, S. Cao, Y. Zhang, L. Ma, and L. Xie, “A transcription prompt-based efficient audio large language model for robust speech recognition,” in *Interspeech 2024*, 2024, pp. 1905–1909.
- [13] L. Cecchi and P. Babkin, “ReportGPT: Human-in-the-loop verifiable table-to-text generation,” in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, F. Dernoncourt, D. Preoțiuc-Pietro, and A. Shimorina, Eds. Miami, Florida, US: Association for Computational Linguistics, Nov. 2024, pp. 529–537. [Online]. Available: <https://aclanthology.org/2024.emnlp-industry.39/>
- [14] K. Qian, Y. Sang, F. Bayat†, A. Belyi, X. Chu, Y. Govind, S. Khorshidi, R. Khot, K. Luna, A. Nikfarjam, X. Qi, F. Wu, X. Zhang, and Y. Li, “APE: Active learning-based tooling for finding informative few-shot examples for LLM-based entity matching,” in *Proceedings of the Fifth Workshop on Data Science with Human-in-the-Loop (DaSH 2024)*, E. Dragut, Y. Li, L. Popa, S. Vucetic, and S. Srivastava, Eds. Mexico City, Mexico: Association for Computational Linguistics, Jun. 2024, pp. 1–3. [Online]. Available: <https://aclanthology.org/2024.dash-1.1/>
- [15] T. Brown *et al.*, “Language models are few-shot learners,” in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 1877–1901. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfc4967418bfb8ac142f64a-Paper.pdf
- [16] M. Del Rio, N. Delworth, R. Westerman, M. Huang, N. Bhandari, J. Palakapilly, Q. McNamara, J. Dong, P. Želasko, and M. Jetté, “Earnings-21: A practical benchmark for asr in the wild,” in *Interspeech 2021*, 2021, pp. 3465–3469.
- [17] OpenAI, “Gpt-4o: A new era of multimodal ai,” 2024, accessed: February 15, 2025. [Online]. Available: <https://openai.com/blog/gpt-4o>
- [18] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, “Qlora: Efficient finetuning of quantized llms,” *Advances in Neural Information Processing Systems*, vol. 36, 2024.
- [19] E. F. Tjong Kim Sang and F. De Meulder, “Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition,” in *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, 2003, pp. 142–147. [Online]. Available: <https://aclanthology.org/W03-0419/>