LLM Selection: Improving ASR Transcript Quality via Zero-Shot Prompting

Grace LeFevre
Northwestern University
Evanston, IL
gracelefevre@u.northwestern.edu

Jordan Hosier, Yu Zhou, Vijay K. Gurbani
Vail Systems, Inc.
Chicago, IL
{jhosier, yzhou, vgurbani}@vailsys.com

***Abstract*—When transcribing telephony audio, Automatic Speech Recognition (ASR) engines often produce noisy output with high word error rates (WER), which impacts the efficacy of downstream analyses that process this transcribed text. We present two experiments demonstrating how an LLM selection method can be used to improve the quality of telephony-based transcripts. This involves generating transcripts for a dataset from multiple ASR engines and prompting an LLM to select the best ASR transcript from the options. Our results indicate that our proposed technique approaches the optimal performance that would be achieved if every transcript was verified against the corresponding ground truth (the Oracle approach). Overall, we find that this method can make notable WER improvements to ASR transcriptions of telephony audio. Further, maximizing performance gain requires utilizing an appropriate targeted improvement strategy.**

I. Introduction

An important challenge in ASR systems is the mismatch of expectations regarding the characteristics of the out-of-distribution audio data, i.e., a Large-Vocabulary Conversational Speech Recognition (LVCSR) system where the audio data for decoding does not correspond to in-distribution audio data used during training. Most commercial and open-source ASR engines are trained on audio data obtained from studio-quality conditions with limited, or no background noise. However, characteristics of the out-of-distribution data are harder to control and have the potential to cause the ASR to produce erroneous transcripts.

For example, a cellular telephone user interacting with an ASR system will exhibit characteristics reflective of the channel, like background noise or jitter, because of poor cellular radio reception. These in turn will cause the ASR system to make different type of errors: homophone substitution ("there" vs. "their"), phoneme substitution ("bat" vs. "pat"), and word boundary ambiguities ("I scream" vs. "ice cream"). Further, poor audio quality can also distort prosodic elements of a speech and make it hard to distinguish fillers and disfluencies ("ah", "umm", etc.).

When such erroneous transcripts generated by such LVCSR systems is fed to back-end applications (like intent determination, call analysis, or sentiment analysis), performance degrades as does the quality of experience of a user interacting with such a system. This underscores the importance of developing effective methods for improving the quality of ASR transcriptions of telephony audio. In this work, we explore one method of using a Large Language Model (LLM) to do so.

Contribution: We present a novel *LLM selection method*, using an LLM to select the best ASR transcript for a document given output of multiple ASR engines. We demonstrate the usefulness of employing multiple ASRs to perform LLM-based ASR transcription improvement and apply this approach to telephony data.

The rest of this paper is structured as follows: Section II positions our work in the context of surveyed literature, Section III describes the datasets and the methodology. Section IV presents our results and discussion, and Section V concludes the paper with a summary of our findings and limitations.

II. Related Work

LLM-based generative error correction and rescoring of an ASR engine's n-best hypothesis list has shown performance improvements over baseline rescoring methods, in some cases surpassing n-best oracle performance [1], [2], [3], [4], [5]. In all, we contribute to and build on surveyed literature in two ways:

1. **Multiple ASR Engines:** Though these approaches have varied in terms of architectures and prompting strategies used, their primary focus has been improving the output of a *single* ASR engine using LLMs. In contrast, we present a test case exploring the possibility of achieving generative LLM-based improvements to ASR output given transcripts produced by *multiple* ASRs. More specifically, we apply a zero-shot, in-context learning approach [6] to the task of selecting the best ASR output from a set of options generated by different ASRs.

2. **Use of *1*-best hypothesis instead of *n*-best:** A notable difference in our work when compared to the reviewed literature is that we do not consider n-best hypotheses, rather, we only consider the final hypothesis (*1*-best) from the ASR engine. We will motivate the reason for this in Section IV-C.

3. **Telephony Domain:** Notably, prior work does not investigate LLM-based ASR error correction for audio generated by 4G and 5G cellular phones, typically relying instead on pre-recorded public datasets like the Airline Travel Information System [7] (ATIS) and Wall Street Journal [8] (WSJ) corpora.

Even large multi-domain ASR datasets like GigaSpeech [9] do not include any telephonic audio.

Transcription accuracy is particularly important for telephony data since results are often used for automated customer service applications. Moreover, the telephony domain is particularly challenging for automatic speech recognition due to possible degradations in audio quality when the radio interface is resource constrained and due to the conversational nature of the speech. Common disruptions to signal transmission include packet loss, delay, and repetitions, which interfere with speech recognition of telephonic audio [10]. Furthermore, speech in the telephony domain is often spontaneous (not read from a script) and informal, including frequent self-corrections and disfluencies [11].

These are all characteristics that make both ASR transcription and LLM error correction more challenging. For example, in some domains LLM-suggested corrections like improving grammatical errors are desirable and improve performance [5]. In the telephony domain, however, such grammatical corrections often may not matter. For example, consider an automated customer service application; such an application is still able to extract intent (pay-card) from a grammatically incorrect transcript ("I want pay my card"). However, if the ASR encounters a substitution error because of noise on the cellular channel ("I want to play my part"), the intent recognition system is unable to extract the correct intent. In the telephony call center domain, accurate transcripts are more important than fixing grammatical errors.

III. Datasets and Methodology

A. Datasets

We use two telephony datasets. Both are drawn from the same context—customers responding to survey questions regarding their recent customer service experience—and consist of short, single-speaker audio files with accompanying human-annotated gold labels. They are in different distributions, with the median gold-label wordcount of the second dataset being significantly shorter than that of the first dataset, as shown in Table I. Due to the relatively small dataset, we chose to use the full datasets for analysis over a train/test split, since there is no need for model training as we are using pre-trained ASRs.

	Dataset #1	**Dataset #2**
documents	911	918
median word count	16	7
total duration (hours)	3.7	1.0
mean (seconds)	14.7	3.7

TABLE I
Data Distribution.

We use a third dataset to motivate our assertion that using *1*-best hypothesis is much more efficient compared to using *n*-best hypotheses. The details of this dataset are provided in Section IV as the dataset is not used in our LLM selection method discussed in this section.

	Dataset #1	**Dataset #2**
Whisper	0.108	0.151
Speechmatics	0.158	0.121
Google telephony	0.121	0.152
Empirical min.	*0.074*	*0.078*

TABLE II
ASR Model performance (WER) on Datasets #1 and #2.

B. ASR Engines

We use three different ASR models in our experiments, generating three sets of transcripts for both of our datasets.

- **Whisper:** the medium sized version of OpenAI's generative Transformer based encoder-decoder ASR model [12].[1]
- **Speechmatics:** a commercial ASR system utilizing the traditional two-component acoustic model and language model.[2]
- **Google telephony:** Google Cloud's speech-to-text model specifically trained for transcribing telephony audio.[3]

The performance of these ASRs on both datasets is shown in Table II. The best-performing ASR model on Dataset #1 is Whisper, achieving a WER of 10.8%. The best-performing ASR model on Dataset #2 is Speechmatics, achieving a WER[4] of 12.1%. This offers further evidence that the two datasets are in different distributions.

C. LLM Used

In this study, we evaluated two LLM models: Meta's Llama-3.1-70B-Instruct[5] 70 billion parameter model and Mixtral 8x7B sparse mixture-of-experts model[6]. Preliminary experimentation revealed that the Llama-3.1-70B-Instruct outperformed Mixtral in terms of transcript improvement, leading to its selection as the LLM for the remainder of this work. Specifically, we ran the Llama-3.1-70B-Instruct-Q6_K_L model, 6-bit quantized, and running on a single NVIDIA H100 with 80 GB VRAM.

D. LLM Selection Method

After obtaining transcriptions from multiple ASR engines for every audio file in a dataset, our LLM selection method consists of the following three steps.

1. **Calculate empirical minimum WER:** Using ground truth labels, analyze comparative ASR performance on the dataset. Calculate the empirical minimum WER to determine whether LLM selection can yield improved performance. The empirical WER for a document is calculated by choosing the transcript from one of the three ASR engines that results in the minimum WER.

[1]Version released on 11.17.2023
[2]https://www.speechmatics.com/
[3]https://cloud.google.com/speech-to-text/docs/transcription-model
[4]The Word Error Rate, or WER, is a widely accepted metric in evaluating ASR models; the lower the WER, the better the model. Please see the appendix for further information on calculating the WER.
[5]https://huggingface.co/meta-llama/Llama-3.1-70B-Instruct
[6]https://mistral.ai/technology/#models

2. **Prompt LLM to select best ASR transcript:** Using information about the dataset domain, the ASR models, and their performances on the dataset, prompt an LLM to choose the ASR transcript most likely to be correct.

3. **Develop targeted improvement strategy:** Calculate the overall WER of the LLM-selected transcripts and analyze which documents in the dataset benefit from the method. This analysis enables selective application of the method to a targeted subset of the data.

Further details on these three steps are provided below.

Empirical Minimum WER: Broadly, our approach resembles an ensemble system where the individual ASR engines can be modeled as independent base classifiers. For such an ensemble to perform optimally, the errors from the base classifiers should not be correlated [13]. To successfully apply an ASR selection method, LLM-based or otherwise, there needs to be evidence that the ASR transcripts have complementary patterns of error. If so, selecting the best transcript for each document can yield better performance overall than any single ASR.

We quantitatively verified this on our datasets by calculating the theoretical best performance that could be achieved on the datasets using an ASR selection approach. Taking the best-performing (lowest WER) ASR transcript for each document in a dataset and calculating their overall WER yields the *empirical minimum WER* that could be achieved via a selection method for that dataset. The empirical minimum WER values for both datasets are shown along with the ASR performances in Table II. For Dataset #1, the empirical minimum WER is .034 lower than the best-performing ASR. For Dataset #2, the difference is even larger at .043 lower than the best-performing ASR. This demonstrates that significant improvement via a selection approach is possible for our datasets.

LLM Prompting: We implemented LLM selection of the best ASR transcript for each document in our datasets by prompting the instruction-tuned Llama3.1-70B model [14]. We present results from the best-performing prompt here (the prompt is shown in Table III[7]). This prompt included domain-specific and distribution-specific information, with details on the (1) customer service context, (2) the ASR model architectures, and (3) their comparative performance on the data. The LLM was instructed to choose the ASR transcription most likely to be true to the original audio file (4). We found this last inclusion particularly essential, as otherwise the LLM was likely to choose the transcript that was most grammatically correct or contained the most standard English.

Targeted Improvement Strategies: Due to the nature of the method, applying it to full datasets carries the risk of performance loss, since the LLM could sometimes select a less correct transcript as the best one. In many use cases, it might be most useful to target method application to only a subset of the data. For this reason, we analyze the documents in a dataset that benefit most from the LLM selection method. This enables the development of targeted improvement strategies that can be quickly tested and implemented on additional datasets. In our results, we discuss two targeted improvement strategies that were effective on our datasets.

[7]Table III shows the final prompt used in Dataset #1; the final prompt used in Dataset #2 was similar with one change: the comparative performance of ASRs on Dataset #2 is different: in Dataset #2, Speechmatics is the best performing ASR engine while Google Telephony is the worst performing.

IV. Results and Discussion

A. Experiment #1

The LLM selection method achieved an overall WER of 0.091 on Dataset #1, an improvement of 0.017 over the best-performing single ASR (Whisper, .108).

For this dataset, ASR disagreement effectively measures transcription quality—the more the three ASR engines disagree about the correct transcription, the less accurate they are all likely to be overall. To measure this, we calculated an ASR disagreement score for each document by summing the three pairwise edit distance scores. This enabled us to examine subsets of the data with higher ASR disagreement.

Table 4 shows the ASR disagreement rate for the documents. For the 20% disagreement partition—i.e., the 20% of the documents with the highest disagreement scores—the WER is the highest. This is expected, as these 20% represents the fraction of documents with the highest disagreement scores from the three ASR engines. As the partition size increases, the WER decreases as documents with lower disagreement scores are brought into the partition. For the 20% disagreement partition, the best single ASR WER is 0.277 (Google Telephony); our method further decreases the WER of this partition by 18.05% to 0.227. For the 50% of the dataset with the most ASR disagreement, the best single WER is 0.175; our method further decreases the WER by 17.71% to 0.144. And finally, when the entire dataset is analyzed, our method decreases the WER by 15.74%, from 0.108 to 0.091.

These results highlight that the documents with the highest ASR disagreement benefit the most from the LLM selection method. This means that focusing on applying the LLM selection method to documents with high ASR disagreement is a promising targeted improvement strategy for this dataset. Since implementing this improvement strategy only requires disagreement between ASR engines, it can be applied to other datasets without ground truth.

B. Experiment #2

The LLM selection method achieved an overall WER of 0.119 on Dataset #2. This represents no significant improvement over the best-performing single ASR on the full dataset (Speechmatics, 0.121).

ASR disagreement was not an effective targeted improvement strategy for Dataset #2. However, some documents in the dataset do benefit from the LLM selection method—specifically, the shortest documents. This is shown in Figure 1, which splits the dataset into three bins based on gold-label word count.

For documents in the shortest bin, the LLM selection method achieves an overall WER of 0.126, a 0.02 improvement over the best-performing single ASR (Google telephony, 0.146). The same pattern does not hold for the other wordcount

```
(1) You are a helpful transcription error correction assistant. I have a telephony dataset
    consisting of customers answering survey questions about their experience speaking to a
    customer service representative.
(2) I transcribed an audio file from this dataset using three Automatic Speech Recognition
    models. The first ASR model is Whisper, a generative transformer-based model. The second
    ASR model is Speechmatics, a traditional ASR that uses an acoustic model and a language
    model. The third ASR model is a Google Cloud model trained to transcribe telephony audio.
(3) Overall, Whisper is the best performing model and Speechmatics is the worst performing
    model, but all three models make mistakes sometimes.
(4) Given the transcriptions produced by these ASR models, your task is to choose which
    transcription you think is most likely to be the correct transcription. A correct
    transcription should be semantically coherent, fit the customer service survey context
    described above, and stick as closely as possible to the content of the original audio
    file. It is likely that all the transcriptions contain inaccuracies, but please choose
    the one you think is most correct. Begin your response with your reasoning and end your
    response with your prediction of the correct transcription. Do not include any additional
    explanation after you state the correct transcription. At the end of your response, please
    put "transcription:" followed by the text of the transcription you selected so I can
    easily extract your prediction from your response.
    Here are the three ASR-generated transcriptions:
    ASR transcription #1: {...}
    ASR transcription #2: {...}
    ASR transcription #3: {...}
```

TABLE III
LLM PROMPT. FINAL ROMPT USED TO IMPLEMENT LLM SELECTION METHOD ON DATASET #1.

	most-disagreeing x%		
	100%	**50%**	**20%**
Whisper	0.108	0.175	0.305
Speechmatics	0.158	0.262	0.395
Google telephony	0.121	0.188	0.277
LLM selection	**0.091**	**0.144**	**0.227**
Empirical min.	*0.074*	*0.120*	*0.187*

TABLE IV
DATASET #1 RESULTS. LLM SELECTION METHOD YIELDED MOST IMPROVEMENT ON DOCUMENTS WITH HIGHEST ASR DISAGREEMENT.

bins, which show either comparable or worsened performance of the LLM selection method compared to the best-performing single ASR. This suggests that wordcount is an appropriate targeted improvement strategy for this dataset; applying the LLM selection method to only the shortest documents yields a performance improvement.

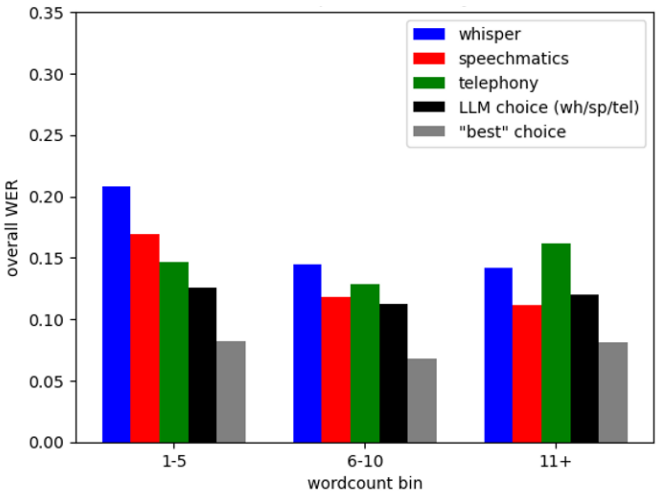

Fig. 1. **Dataset 2 Results.** Shorter documents benefit most from the LLM selection method.

C. 1-best vs. n-best hypotheses

Our assertion that *1*-best hypothesis is superior to *n*-best is empirically drawn from the open-source Kaldi ASR toolkit [15]. The base Kaldi source code contains a method called *SentenceLevelConfidence()* that returns float value signifying the difference between the “best" sentence and the “second-best" sentence. The best and second-best sentences represent the best and second-best paths in the decoding lattice. The return value of the function is one of three values: a positive number that represents the difference between the best path and second-best path, 0 if there were no paths in the lattice, or ∞ (infinity) if there was only one path in the lattice.

The value returned from the function appears to represent a difference — or a distance measure — between the best and second-best path; thus lower values of the return value would indicate shorter distances, i.e., the best and second-best paths are approximately the same. Therefore, a reasonable hypothesis would be that if the best and second-best paths are approximately similar according to that distance measure, the confidence that the sentence has been correctly decoded should be high. The following analysis explores the validity of this hypothesis using a dataset shown in Table V. The dataset is also drawn from the telephony domain, and mean of the

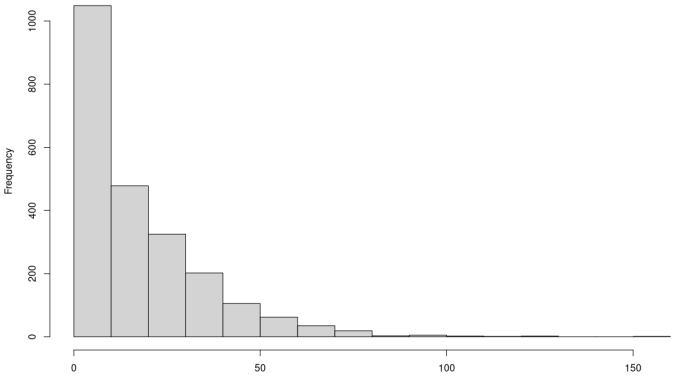

Fig. 2. ***SentenceLevelConfidence()* distribution.**

dataset is between the means of Dataset #1 and Dataset #2 as shown in Table I.

	Statistic
Observations	3081
Total duration	4 hours 47 mins
Range	0.7-8.1 seconds
Mean	5.6 seconds

TABLE V
DATASET FOR *1*-BEST HYPOTHESIS DETERMINATION.

Of the 3,081 observations in the test dataset, 792 observations resulted in the *SentenceLevelConfidence()* method returning infinity. That is, there was only one path in the lattice, meaning the likelihood that this is the best path (most correct transcript) would be very high (∞).These 792 observations cannot be improved upon so we exclude these observations from our analysis. The remaining observations return values ranging from 0 to 151.84 (the x-axis in Figure 2).

The mass of the histogram is between values of 0-40. However, there is a long tail that is not readily visible in the figure. Based on the right skewness of the histogram, a cut-off point can be established on the x-axis that represents a high confidence of the decoded sentence: any sentence that had a return value less than this threshold would be considered to be decoded with a high confidence. Further analysis is required to evaluate the appropriateness of such a cut-off point, as described next.

The evaluation of the goodness of a cutoff point involves the interplay between two variables: the sentence-level WER (α) of an observation, and its return value from *SentenceLevelConfidence()* API (β). There should be a positive correlation between α and β: when the WER (α) of the sentence is low, the confidence (β) — i.e., the value returned from the *SentenceLevelConfidence()* API — should also be low. (Recall that lower return values from the *SentenceLevelConfidence()* API imply less disagreement between the first and second-best sentences.)

We will allow α to be the independent variable and set it to a value of 0.13, close to the 50% value of LLM selection in Table IV. 1,514 observations have values of $\alpha \leq 0.13$; their distribution is shown in table below. Table VI demonstrates

	Statistic
Observations	1514
Mean	19.85
Std. Dev.	17.96
Median	15.45
Minimum	0
Maximum	151.84

TABLE VI
DISTRIBUTION OF β FOR $\alpha \leq$ 0.13.

that the range of β includes both extremes: the minimum value of 0 and the maximum value of 151.84. Effectively, this implies α is not correlated with β (r = -0.07), and therefore the *SentenceLevelConfidence()* return values should not be used as a measure of sentence-level confidence. Based on this analysis, we reach the conclusion that the return value from this function is an unstable measure of sentence-level confidence.

While this analysis is Kaldi-specific, confidence metrics are common return value in ASR systems. Such metrics should be closely evaluated for accuracy and appropriateness prior to use in such tools.

V. CONCLUSION

Taken together, these experiments suggest that ASR transcriptions of telephony audio can be improved via the LLM selection method described here. We presented two targeted improvement strategies that were effective on our datasets. This method can be implemented on additional datasets given knowledge of their domain and distribution, as well as ground truth labels for a small subset.

LIMITATIONS

This work provides a proof-of-concept for the LLM selection method we present, tested on two proprietary datasets. Further testing on additional larger datasets (including open-source datasets) is needed to understand the full utility of this approach. Moreover, it would be useful for future work to include further testing of possible LLM prompts, including an ablation analysis.

The discussion presented in Section IV-C is conducted on the best-of-breed open source Kaldi ASR engine [15]. Kaldi uses time-delay neural networks, which are feed-forward architectures that use acoustic features as input and learn from an increasingly sparse context window at each layer. Whisper, by contrast, is architected as an encoder-decoder Transformer network [12], while Google telephony combines convolutions and transformers for speech recognition [16]. Published literature on Speechmatics' architecture is not readily available; based on derivative works that have used Speechmatics (Williams et al. [17]), it appears that Speechmatics uses a recurrent neural network language model.

Given the differences in architectures enumerated in the above paragraph, it is fair to contemplate whether the empirical results we observe with Kaldi in Section IV-C hold across other ASR architectures? We believe that the answer

is yes, because Kaldi's modular approach allows developers to fine-tune each stage (feature extraction, acoustic modeling, language modeling, and decoding); the result of this is an optimized, single-purpose model. Whisper, by contrast, is a multi-purpose model that uses a sequence-to-sequence approach where the decoder generates text tokens directly from the encoder's representations using a supervised and semi-supervised learning approach. Under such architectures, there is a tendency to *hallucinate*, defined as "undesirable generated text "that is nonsensical, or unfaithful to the provided source input" [18]. Koenecke et al. [19] report that up to 1% of Whisper's transcripts contained entire made-up sentences that did not occur in the corresponding audio file. They further quantify that nearly 40% of hallucinations are harmful (as opposed to harmless and random). We plan to investigate different architectures in future work to authoritatively determine whether the behaviour of a single purpose, optimized ASR model like Kaldi can serve to inform the community about the need to consult an *n*-best hypotheses set or use the *1*-best hypothesis.

Appendix: Word Error Rate

The WER [20] is a widely accepted standard measure of ASR performance; it is expressed as a value between [0, 1.0] or as a percentage. ASR systems seek to minimize the WER. It is represented as the ratio of the number of edits required to transform a hypothesis string into a reference string to the total number of words in the reference string, or

$$\text{WER} = \frac{I + D + S}{N} \tag{1}$$

where S = number of substitutions required to change the hypothesis string to the reference string, D = number of deletions required, I = number of insertions, and N = total number of words in the reference string. Lower values of WER are preferred since they indicate an ASR model that makes less errors.

Before the WER is calculated, the reference and hypothesis strings are aligned using alignment algorithms [21]. The following example demonstrates the WER calculation after alignment has taken place.

```
REF: I want to pay  my card.
HYP: I **** to play my part.

Insertions: 0
Deletions:  1 (want)
Substitutions: 2 (play, part)
WER = (2 + 1 + 0) / 6 = 0.50
```

References

[1] J. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (rover)," in *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*. IEEE, 1997, pp. 347–354.

[2] R. Ma, M. Qian, P. Manakul, M. Gales, and K. Knill, "Can generative large language models perform asr error correction?" *arXiv preprint arXiv:2307.04172*, 2023.

[3] C.-H. H. Yang, Y. Gu *et al.*, "Generative speech recognition error correction with large language models and task-activating prompting," in *ASRU 2023*, 2023. [Online]. Available: https://www.amazon.science/publications/generative-speech-recognition-error-correction-with-large-language\-models-and-task-activating-prompting

[4] C. Chen, Y. Hu *et al.*, "Hyporadise: An open baseline for generative speech recognition with large language models," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[5] S. Radhakrishnan, C.-H. Yang *et al.*, "Whispering LLaMA: A cross-modal generative error correction framework for speech recognition," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 10 007–10 016. [Online]. Available: https://aclanthology.org/2023.emnlp-main.618

[6] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *Advances in neural information processing systems*, vol. 35, pp. 24 824–24 837, 2022.

[7] C. T. Hemphill, J. J. Godfrey, and G. R. Doddington, "The ATIS spoken language systems pilot corpus," in *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27,1990*, 1990. [Online]. Available: https://aclanthology.org/H90-1021

[8] D. B. Paul and J. M. Baker, "The design for the Wall Street Journal-based CSR corpus," in *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992*, 1992. [Online]. Available: https://aclanthology.org/H92-1073

[9] G. Chen, S. Chai *et al.*, "Gigaspeech: An evolving, multi-domain asr corpus with 10,000 hours of transcribed audio," *arXiv preprint arXiv:2106.06909*, 2021.

[10] J. Bochner, M. Indelicato, and P. Konnur, "Effects of sound quality on the accuracy of telephone captions produced by automatic speech recognition: A preliminary investigation," *American Journal of Audiology*, vol. 32, no. 1, pp. 243–250, 2023.

[11] W. Xiong, J. Droppo *et al.*, "Achieving human parity in conversational speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. PP, 10 2016.

[12] A. Radford, J. W. Kim *et al.*, "Robust speech recognition via large-scale weak supervision," in *International conference on machine learning*. PMLR, 2023, pp. 28 492–28 518.

[13] P.-N. Tan, M. Steinbach, A. Karpatne, and V. Kumar, *Introduction to Data Mining (2nd Edition)*, 2nd ed. Pearson, 2018.

[14] A. Dubey, A. Jauhri *et al.*, "The llama 3 herd of models," *arXiv preprint arXiv:2407.21783*, 2024.

[15] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. K. Goel, M. Hannemann, P. Motlícek, Y. Qian, P. Schwarz, J. Silovský, G. Stemmer, and K. Veselý, "The kaldi speech recognition toolkit," in *IEEE Workshop on Automatic Speech Recognition and Understanding*, 2011.

[16] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-augmented transformer for speech recognition," in *Interspeech*, 2020, pp. 5036–5040.

[17] W. Williams, N. Prasad, D. Mrva, T. Ash, and T. Robinson, "Scaling recurrent neural network language models," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5391–5395.

[18] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, and P. Fung, "Survey of hallucination in natural language generation," *ACM Comput. Surv.*, vol. 55, no. 12, Mar. 2023. [Online]. Available: https://doi.org/10.1145/3571730

[19] A. Koenecke, A. S. G. Choi, K. X. Mei, H. Schellmann, and M. Sloane, "Careless whisper: Speech-to-text hallucination harms," in *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency*, ser. FAccT '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 1672–1681. [Online]. Available: https://doi.org/10.1145/3630106.3658996

[20] K. Zechner and A. Waibel, "Minimizing word error rate in textual summaries of spoken language," in *1st Meeting of the North American Chapter of the Association for Computational Linguistics*, 2000.

[21] A. C. Morris, V. Maier, and P. D. Green, "From WER and RIL to MER and WIL: improved evaluation measures for connected speech recognition." in *Interspeech*, 2004, pp. 2765–2768.